

Oktober 2008

Computeralgebra

Rundbrief

GI_DMV_GAMM

- ▶ Tagungen der Fachgruppe in Soest, Kassel und München
- ▶ Imaginary-Ausstellung im Jahr der Mathematik
- ▶ Neues über Systeme: GAP, Magma, Maple
- ▶ CA in der Schule: Schulversuch CALiMERO



Inhaltsverzeichnis

Inhalt	3
Impressum	4
Mitteilungen der Sprecher	5
Tagungen der Fachgruppe	7
Aktivitäten zum Jahr der Mathematik	9
<i>Algebraische Geometrie im Jahr der Mathematik 2008</i> (Gert-Martin Greuel, Andreas Daniel Matt)	10
Themen und Anwendungen der Computeralgebra	13
<i>From the Lagrangian to Feynman Rules with FeynRules</i> (Neil D. Christensen, Claude Duhr)	13
<i>Counting points on curves over finite fields</i> (Frederik Vercauteren)	16
Neues über Systeme	19
<i>Galoisgruppen in Magma</i> (Claus Fieker, Jürgen Klüners)	19
<i>Vorstellung des GAP-Projektes</i> (Frank Lübeck)	21
<i>Neues aus Waterloo: Maple 12 und MapleSim 1.0</i> (Thomas Richard)	23
Computeralgebra in der Schule	26
<i>Der Schulversuch CALiMERO</i> (Henning Körner)	26
Computeralgebra in der Lehre	31
<i>Integration der Nutzung eines CAS in der Veranstaltung „Elementare Analysis“</i> (Frauke Arndt, Hans-Wolfgang Henn)	31
Publikationen über Computeralgebra	36
Besprechungen zu Büchern der Computeralgebra	36
<i>Greuel, Pfister: A Singular Introduction to Commutative Algebra (Franz Pauer)</i>	36
<i>Levandovskyy (Ed.): Non-Commutative Gröbner Bases and Applications (Werner M. Seiler)</i>	37
<i>Rosenkranz, Wang (Eds.): Gröbner Bases in Symbolic Analysis (Wilhelm Plesken)</i>	37
<i>Ziegenbalg: Algorithmen von Hammurapi bis Gödel (Elkedagmar Heinrich)</i>	38
Berichte von Konferenzen	39
Hinweise auf Konferenzen	42
Fachgruppenleitung Computeralgebra 2008-2011	47

Impressum

Der Computeralgebra-Rundbrief wird herausgegeben von der Fachgruppe Computeralgebra der GI, DMV und GAMM (verantwortlicher Redakteur: Dr. Markus Wessler, markus.wessler@hm.edu).

Der Computeralgebra-Rundbrief erscheint halbjährlich, Redaktionsschluss 28.02. und 30.09. ISSN 0933-5994. Mitglieder der Fachgruppe Computeralgebra erhalten je ein Exemplar dieses Rundbriefs im Rahmen ihrer Mitgliedschaft. Fachgruppe Computeralgebra im Internet:
<http://www.fachgruppe-computeralgebra.de>.

Konferenzankündigungen, Mitteilungen, einzurichtende Links, Manuskripte und Anzeigenwünsche bitte an den verantwortlichen Redakteur.

Die Geschäftsstellen der drei Trägergesellschaften:

GI (Gesellschaft für Informatik e.V.)
Wissenschaftszentrum
Ahrstr. 45
53175 Bonn
Telefon 0228-302-145
Telefax 0228-302-167
gs@gi-ev.de
<http://www.gi-ev.de>



DMV (Deutsche Mathematiker-Vereinigung e.V.)
Mohrenstraße 39
10117 Berlin
Telefon 030-20377-306
Telefax 030-20377-307
dmv@wias-berlin.de
<http://www.dmv.mathematik.de>



GAMM (Gesellschaft für Angewandte Mathematik und Mechanik e.V.)
Technische Universität Dresden
Institut für Festkörpermechanik
01062 Dresden
Telefon 0351-463-33448
Telefax 0351-463-37061
GAMM@mailbox.tu-dresden.de
<http://www.gamm-ev.de>



Mitteilungen der Sprecher

Liebe Mitglieder der Fachgruppe Computeralgebra,

am 20. September 2008 fand in Erlangen nach Abschluss der DMV-Tagung 2008 die zweite Sitzung der Fachgruppenleitung statt. In einer sehr interessanten Sitzung wurde über unsere vielfältigen Aktivitäten berichtet und diskutiert, und es wurden die Weichen für weitere Aktivitäten gestellt.



Ein Teil der Fachgruppenleitung am Vorabend der Sitzung in Erlangen

Unser Sonderheft des Computeralgebra-Rundbriefs zum Jahr der Mathematik, das wir im April 2008 an alle Gymnasien versandt haben, ist ein großer Erfolg! Die Resonanz ist vielfältig und durchwegs positiv. Das Heft kann von der Internetseite <http://www.fachgruppe-computeralgebra.de/JdM> heruntergeladen werden. Ferner können weitere gedruckte Hefte beim Sprecher angefordert werden. Details hierzu finden sich auf der obigen Internetseite.

Der Schülerwettbewerb, den wir in unserem Sonderheft ausgeschrieben haben, ist als Event des „Jahrs der Mathematik“ auf der Website <http://www.jahr-der-mathematik.de> und im dort ebenfalls verfügbaren Jahresprogramm zu finden und macht unsere Fachgruppe zum offiziellen Partner des Wissenschaftsjahres. Wettbewerbsbeiträge können bis zum 31. Oktober eingereicht werden, die Preisverleihung findet dann am 13. Dezember in Passau statt. Weitere Informationen hierzu gibt es auf S. 9.



Schreiben Sie an einer Facharbeit oder Studienarbeit zum Thema Computeralgebra? Haben Sie Lust, sich mit computerorientierter Mathematik zu beschäftigen? Dann nehmen Sie doch teil am

Computeralgebra-Wettbewerb

Anmeldung bis zum 15.10.2008 unter WCA@mathematik.uni-kassel.de
Einsendeschluss: 31.10.2008
Siegerehrung: 13.12.2008 in Passau
Teilnahmeformular, Wettbewerbsbedingungen und Themenvorschläge auf der Webseite
www.fachgruppe-computeralgebra.de/JdM/

1. Preis:	2 ¹⁰ €
2. Preis:	2 ⁹ €
3. Preis:	2 ⁸ €
4. Preis:	2 ⁷ €
5. Preis:	2 ⁶ €





Prof. Dr. Ernst W. Mayr bei seiner Präsentation in Hagenberg

Auch unsere Tagung Computeralgebra in Lehre, Ausbildung und Weiterbildung: Computeralgebra und ihre Didaktik – Einfluss auf Lernen und Prüfen, welche in der Woche nach Ostern in Soest stattfand, war offizieller Teil des Jahr-der-Mathematik-Programms. Einen Bericht über die Tagung in Soest finden Sie auf S. 7.

Vom 14.-16. Mai 2009 wird die Fachgruppe wieder eine Tagung zur Forschung in der Computeralgebra veranstalten, die in Kassel stattfinden wird. Und unsere Reihe zum Einsatz von Computeralgebra in Lehre und Didaktik soll 2010 in der Woche nach Ostern weitergeführt werden. Ferner unterstützt die Fachgruppe die Tagung CASK 2009 (<http://www.cask.fh-konstanz.de>) über den Computeralgebraeinsatz an Fachhochschulen, welche am 12. und 13. März 2009 in Konstanz stattfinden wird.

Im Sommer 2010 schließlich wird die Fachgruppe die internationale Tagung ISSAC 2010 in München organisieren. Dies beschlossen die Tagungsteilnehmer der diesjährigen ISSAC-Tagung mit großer Mehrheit in Linz. Ernst W. Mayr von der TU München als lokaler Organisator und Wolfram Koepf als Sprecher der Fachgruppe hatten unsere Bewerbung in Konkurrenz zu einer Bewerbung aus Boston präsentiert.

Weiter wurde auf dem ISSAC Business Meeting in Linz die stellvertretende Sprecherin unserer Fachgruppe Elkedagmar Heinrich als Nachfolgerin von Gerhard Hiß für ein Jahr in das ISSAC Steering Committee berufen.



Der Vortragssaal der ISSAC-Tagung in Hagenberg

Der Richard-Jenks-Software-Preis (www.sigsam.org/awards/jenks) in Höhe von 1.000 € ging diesmal an das GAP-System und damit wiederholt an ein originär deutsches System. In Empfang nahmen das Preisgeld Steve Linton (St. Andrews) und Frank Lübeck aus Aachen, s. auch den Artikel über GAP auf S. 21. Bisherige Preisträger waren Singular (2004) und John Cannon für Magma (2006). Einen ausführlichen Bericht über die ISSAC-Tagung 2008 finden Sie auf S. 40.

Der Sprecher der Fachgruppe, Wolfram Koepf, wurde von den Mitgliedern der DMV als Verantwortlicher für www.mathematik.de mit Wirkung vom Januar 2009 ins Präsidium der DMV gewählt. Wünsche der Fachgruppe bzgl. der Präsenz auf DMV-Tagungen u. ä. können nun noch unkomplizierter eingebracht werden.

Die nächste Sitzung der Fachgruppenleitung findet am 7. Februar 2009 in Kassel statt. Wir hoffen, Sie mit dem vorliegenden Heft wieder gut zu informieren.

Wolfram Koepf

Elkedagmar Heinrich

Tagungen der Fachgruppe

Computeralgebra in Lehre, Ausbildung und Weiterbildung VI: Computeralgebra und ihre Didaktik – Einfluss auf Lernen und Prüfen, 27. – 28.03.2008, Landesinstitut für Schule/Qualitätsagentur Soest

Am 27.-28.3.2008 fand in Soest die 6. Tagung zu CLAW (Computeralgebra in Lehre, Ausbildung und Weiterbildung) mit 24 Teilnehmern unter Leitung von Hans-Wolfgang Henn (TU Dortmund) statt.



Tagungsfoto

Diskutiert wurden die folgenden acht Vorträge.

Frauke Arndt (Technische Universität Dortmund): CAS kennen lernen.

Es handelt sich um einen Bericht über diesbezügliche Veranstaltungen an der Universität Dortmund für künftige Gymnasiallehrer. Zwei unterschiedliche Zugänge wurden erprobt: (i) Kennenlernen der Funktionsweisen von CAS; anschließend (einfache) Transferaufgaben bzw. (ii) Beginn mit komplexem Modellierungs-Beispiel, bei der sich die Studierenden die Grundlagen des CAS selbst erarbeiten mussten.

Beide Zugänge wurden anhand eines Kriterienkatalogs diskutiert. Dabei ergaben sich Vorteile für den ersten Weg, auch wenn dort einige Charakteristika eher oberflächlich realisiert wurden (dies bezieht sich vor allem auf die Einsicht in die Notwendigkeit des Werkzeugs).

Bei der Diskussion wurde darauf hingewiesen, dass das Kennenlernen eines CAS in Schule, Ausbildung und Fortbildung jeweils eine ganz andere Struktur haben muss.

Wolfgang Kroll (Universität Marburg): Überlegungen zu einer Didaktik des CAS-Rechnereinsatzes unter besonderer Berücksichtigung der LK-Abiturprüfungsaufgaben 2007 NRW

Im Mittelpunkt von (Unterricht und) Prüfungen steht nicht das Beherrschen eines Kalküls, sondern das Aufstellen von Ansätzen und das Arbeiten mit sowie

die Interpretation von Ergebnissen der Formelanwendungen. Aufgaben sollten so mit freien Variablen versehen werden, dass inhaltlich sinnvolle Erweiterungen möglich sind. Daher müssen auch die Unterrichtsinhalte reichhaltiger werden. Probleme der Dokumentation bei Arbeit mit CAS wurden angesprochen. Defizite im CAS-Einsatz innerhalb der Stochastik sind noch vorhanden.

Gilbert Greefrath & Andreas Pallack (Pädagogische Hochschule Karlsruhe & Landesinstitut für Schule / Qualitätsagentur Soest): Gute Abituraufgaben – (ob) mit oder ohne Neue Medien

Im Mittelpunkt der Ausführungen stand die provokative These:

Zur Überprüfung von mathematischen Kompetenzen, die im Unterricht mit digitalen Werkzeugen erworben wurden, bedarf es in der Prüfungssituation nicht unbedingt der digitalen Werkzeuge.

Da das Prüfungsformat den vorhergehenden Unterricht mit bestimmt, muss der Einsatz digitaler Werkzeuge im Unterricht als Voraussetzung für die Prüfung sinnvoll bzw. unabdingbar sein. Diese Problematik ist von dynamischer Geometrie-Software schon bekannt. Die These wurde kontrovers diskutiert.

Einerseits: Durch das Verbot von CAS werden den Schülern in Prüfungen produktive Möglichkeiten beschnitten. Es gibt zudem begriffliche Werkzeuge, die man in Prüfungen auch nicht verbieten wird. Dazu passt Dörflers These: Werkzeuge unterstützen nicht nur Kognition, sie sind auch Teil der Kognition.

Andererseits: Es scheint „gute CAS-Aufgaben“ für Prüfungen gar nicht zu geben. Zwar werden manche Begriffe nur durch Werkzeuge gebildet, aber die Begriffe verschwinden nicht, wenn das Werkzeug weggenommen wird. Unstrittig war, dass man sich bei Modellierungsaufgaben auf Teilprozesse beschränken sollte.

Eberhard Lehmann (Berlin): Nachhaltige Konzepte für den MU mit Computern (insbesondere mit CAS)

Grundlegend ist neue (offenere) Unterrichtskultur; Umgang mit Rechnern sollte normal sein. Schwerpunkt ist Problemlösung und Teamarbeit mit vielen Bereichen selbständigen Arbeitens. Das Wechselspiel verbal/symbolisch/tabellarisch/graphisch wird immer wieder verfolgt.

Eine der Hauptkomponenten des Rechnereinsatzes besteht in Visualisierung/Animation, da hierdurch das Verständnis mächtig gefördert wird. Modularisiertes Arbeiten wird durch CAS sehr unterstützt; häufig lohnt sich die nähere Untersuchung solcher Bausteine. Insbesondere ist auch der Transfer fertiger Bausteine auf neue Funktionsklassen möglich, sinnvoll und erhellend.

Henning Körner (Studienseminar Oldenburg): Welche Kompetenzen werden durch CAS-Gebrauch gefördert?

Der Vortrag lieferte viele und sehr überzeugende Beispiele dafür, dass das Denken durch den Rechner nicht ausgeschaltet, sondern wesentlich unterstützt wird. Es wurde deutlich, dass ein Kompetenzaufbau im Wechselspiel aus „zu Fuß“ und „mit CAS“ gelingen kann. Die Frage nach „dem Dahinterstehenden“ wird nicht mehr durch Kalküle verdeckt.

Ebenfalls wurde deutlich, wie Strukturelemente im Verwendungskontext entstehen können. Nicht der Rechner an sich ist mächtig, sondern mächtig wird er in der Hand des Schülers in Kooperation mit dem Lehrer. Bei der Diskussion wurde hervorgehoben, dass in Bezug auf LGS eine begriffliche Progression zu beobachten ist: Manche Schüler arbeiten in den Gleichungen, andere mit den Gleichungen, noch andere nur mit den Namen der Gleichungen. Eine solche Entwicklung ist ohne CAS viel schwerer zu erreichen.

Ernestine Dittrich (Technische Universität Karlsruhe): Maple – 15 Jahre Einsatz in der Schule – Ein Grund zum Feiern?

Es handelte sich um einen Bericht über die bisherigen Erfahrungen in Baden-Württemberg. Deutlich wurde der entscheidende Einfluss der Software auf die Art der Aufgaben und auf die Methode des Unterrichts. Die Vorgabe, dass alle Schüler (mit bzw. ohne CAS) im Zentralabitur die gleichen Aufgaben bekommen sollten, erwies sich als recht realitätsfern.

Bei der Diskussion wurde hervorgehoben, dass CAS-Schüler keineswegs durch mehr Inhalte „bestraft“ werden sollten, sondern die Intensivierung sollte auf die Bereiche Modellieren, Aufstellen eines Ansatzes und Interpretieren gelegt werden.

Reinhard Oldenburg (Universität Frankfurt): Neue Wege in alte Sackgassen

Aus der Beobachtung, wonach händische Fertigkeiten zurückgehen, die Schüler aber zu wenig neue Verhaltensweisen lernen, liegt der Schluss nahe: Mit der Technologie neue Wege zu alten Zielen gehen zu wollen, scheint gescheitert zu sein, auch deswegen, weil manche alten Ziele nicht mehr attraktiv sind. Es wurde die These, Mathematikdidaktik sei technikfeindlich, erläutert und diskutiert: Das Gymnasium zielt mehr auf Wissen und Erkennen; nicht darauf, dass man etwas macht. Technische Aspekte werden kaum als Chance betrachtet, die Struktur der Gegenstände zu klären, sondern eher als lästiges Übel angesehen. In Zusammenhang damit steht die These, dass Substitution eine wichtige algebraische Kernkompetenz ist, dass sie in-

nerhalb der Algebra sogar eine fundamentale Idee bzw. sogar ein semantisches Werkzeug darstellt. Substitutionsübungen sind daher nicht (nur) technischer Art, sondern befördern auch den semantischen Aspekt. Sie sind auch ein gutes Beispiel für den allgemeinen Sachverhalt, dass der Computer kaum neue Methoden generiert, sondern die Reichweite bekannter Methoden gewaltig vergrößert. Ein sinnvoller Computereinsatz erfordert ein mentales Modell der Maschine.

Dass die neuen Wege auch in Sackgassen geführt zu haben scheinen, wird auch daran deutlich, dass die weitaus meisten Studenten, die in ihrer Schulzeit Computereinsatz erlebt hatten, kaum in der Lage waren, die Bedeutung des Computers für mathematische Fragen angemessen darzustellen.

Beim Lösen von Problemen ist eine Strukturierung durch Begriffe hilfreich. Begriffe sollen denkökonomisch sein. Gegenüber technologischem Wandel sind weder Begriffe noch Aufgabenformate invariant! Computerorientierte Mathematik sollte zu neuen Aufgabenformaten führen, so dass die Schüler sehen, dass sich der Technologieeinsatz wirklich lohnt.

Dörte Haftendorn (Universität Lüneburg): „Mathematik für alle“ – Vorlesung mit Einsatz von Computerwerkzeugen: Ein Wagnis mit Erfolg

Es ging um das ehrgeizige (und mit gutem Erfolg erreichte) Ziel, 1200 Studenten (mit einem mathematikfernen Studienschwerpunkt) in 7 Wochen mit 4 Semesterwochenstunden eine Vorstellung von Sinn und Wesen der Mathematik nahezubringen. Inhaltlich sollte deutlich werden, wie die Mathematik hilft, die Welt zu strukturieren; dazu sollten qualitative Denkprozesse initiiert werden.

Thematisch wurde der Bogen von Kryptographie und Graphentheorie bis zu Fragen der Optimierung und Numerik gespannt; Näheres findet man in www.leuphana.de/matheomnibus. Methodisch handelte es sich um eine Vorlesung mit integrierten Übungen (jeweils 4-7 Minuten; Anregung der Kommunikation in Vierergruppen), die von 5 studentischen Hilfskräften unterstützt wurde. Besondere Mühe wurde verwendet bei der Berücksichtigung affektiver Momente (wie Angst), deren Ignorierung immer kontraproduktiv ist. Deutlich wurde, dass eine interaktive Visualisierung (Geogebra, MuPad) in der Analysis eine enorm einsichtsfördernde Kraft hat.

Die Tagung wurde freundlicher Weise von den Firmen Additive, Casio, Scientific, Sciface und Texas Instruments unterstützt.

Jörg Meyer (Hameln)

Aktivitäten zum Jahr der Mathematik

Anlässlich des Jahrs der Mathematik richtet die Fachgruppe einen **Computeralgebra-Wettbewerb** aus. Gesucht sind Facharbeiten, Seminararbeiten o. Ä. zu Themen aus der Computeralgebra. Die genauen Teilnahmebedingungen stehen auf der Webseite <http://www.fachgruppe-computeralgebra.de/JdM/>

Die Anmeldefrist für den Wettbewerb ist bis zum **15.10.2008** verlängert worden. Anmelden können sich interessierte Schüler oder Schülergruppen per E-Mail an die Adresse WCA@fachgruppe-computeralgebra.de. Der Einsendeschluss für die Wettbewerbsbeiträge ist der 31.10.2008. Es winken Geld- und Sachpreise im Wert von insgesamt über 2000 Euro sowie eine Reise zur Siegerehrung nach Passau.

Die Siegerehrung findet am **Samstag, den 13.12.2008**, im Rahmen des Tags der Mathematik an

der Universität Passau statt. Dort wird es (auch als regionale Lehrerfortbildung) Vorträge und Computerübungen zum Thema „Computeralgebra an der Schule“ geben. Ferner ist die Ausstellung „Imaginary 2008“ vom 05.12. bis zum 19.12. in Passau zu Gast, vgl. <http://www.imaginary2008.de> sowie den folgenden Beitrag. Ideen für Beiträge zum Computeralgebra-Wettbewerb findet man auch im Sonderheft dieses Rundbriefs, das die Fachgruppe im Frühjahr herausgab. Extraexamplare dieses Sonderhefts können per E-Mail beim Sprecher der Fachgruppe, Herrn Prof. Dr. Wolfram Koepf, bestellt werden. Ein Versandkostenbeitrag von 7,90 € ist dazu auf das auf der obigen Webseite angegebene Konto zu überweisen.

Martin Kreuzer (Passau)



GI (Gesellschaft für Informatik e.V.)
Wissenschaftszentrum
Altestraße 45
53175 Bonn
<http://www.gi-ev.de>



DMV (Deutsche Mathematiker-Vereinigung e.V.)
Mohrenstraße 39
10117 Berlin
<http://www.dmv.mathematik.de>



GAMM (Gesellschaft für angewandte Mathematik und Mechanik e.V.)
Technische Universität Dresden
Institut für Festkörpermechanik
01062 Dresden
<http://www.gamm-ev.de>



Algebraische Geometrie im Jahr der Mathematik 2008 Eine Zwischenbilanz der Ausstellung IMAGINARY

Gert-Martin Greuel
Mathematisches Forschungsinstitut Oberwolfach

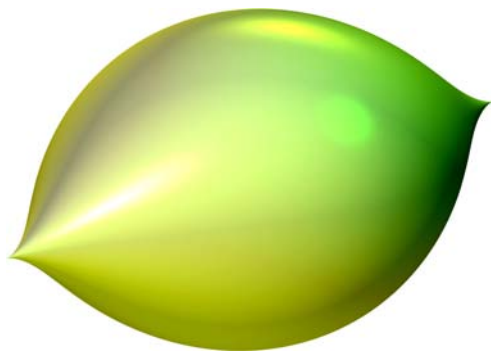
Andreas Daniel Matt
Mathematisches Forschungsinstitut Oberwolfach

greuel@mfo.de
matt@mfo.de



Zusammenfassung

Im Jahr der Mathematik 2008 präsentiert die Wanderausstellung IMAGINARY des Mathematischen Forschungsinstituts Oberwolfach das Gebiet der algebraischen Geometrie durch interaktive Installationen, ansprechende Bildgalerien, Medienarbeit und Wettbewerbe. Das dafür entwickelte Programm SURFER zur schnellen Visualisierung reeller algebraischer Flächen begeistert ein breites Publikum. Wir ziehen eine erste Zwischenbilanz der Ausstellung und teilen unsere Erfahrungen bei der Vermittlung algebraischer Geometrie mit.



Das Logo der Ausstellung: Zitrus $x^2 + z^2 = y^3(1 - y)^3$

Die Idee

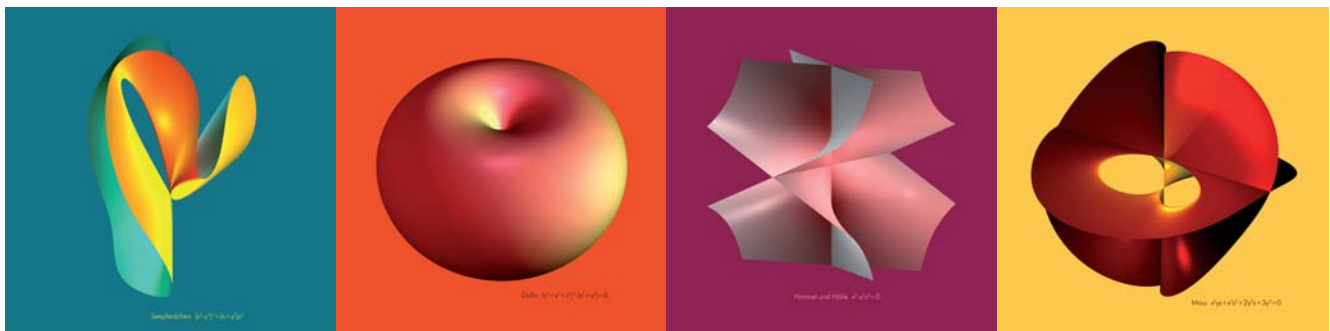
„Es ist die Freude an der Gestalt, die den Geometer ausmacht“. Dieser Satz des deutschen Mathematikers Alfred Clebsch (1833–1872) ist das Motto der Ausstellung. Die Freude an der Gestalt und zusätzlich am eigenen Gestalten soll in der Wanderausstellung „IMAGINARY – Mit den Augen der Mathematik“ vermittelt

werden. Diese interaktive Ausstellung zur algebraischen Geometrie und zu mathematischen Visualisierungen aus anderen Gebieten der Mathematik wird im Jahr der Mathematik 2008 in über 13 deutschen Städten zu sehen sein.

Das Seepferdchen und seine Formel

Prof. Herwig Hauser von der Universität Wien erzeugte aus einfachen algebraischen Gleichungen ästhetisch anspruchsvolle Bilder, gab ihnen interessante Namen und visualisierte sie mit ausgesuchten Farben. Eine Galerie mit Bildern wie Seepferdchen, Dullo, Himmel&Hölle oder Miau lädt die BesucherInnen ein, die Ästhetik und Schönheit der „Gleichungen“ zu bewundern.

Zu jeder Fläche gibt eine Erklärungstafel Einblicke in die mathematischen Eigenschaften und die Erstellung des Bildes. So werden die wichtigen Elemente der Bilder, wie z. B. die Singularitäten, beschrieben.



Algebraische Flächenkunst – 12 Motive werden auch als Posterset angeboten

Persönliche Vermittlung

An jedem Ausstellungsort gibt ein Team von BetreuerInnen, meist StudentInnen oder Dozenten der lokalen Universitäten, eine Einführung in den Zusammenhang zwischen Form und Formel, zwischen Algebra und Geometrie, und steht während der Öffnungszeiten den Besuchern für Fragen und weitergehende Erklärungen zur Verfügung. Das Team wird jeweils vor der Ausstellung durch schriftliches Material und durch eine persönliche Einführung auf die Aufgabe vorbereitet. Es hat sich gezeigt, dass diese individuelle Betreuung der BesucherInnen sehr wichtig ist, einerseits um Vorurteile zur „unverständlichen“ Mathematik abzubauen aber auch um das durch die Ausstellung angeregte Wissensbedürfnis zumindest teilweise zu befriedigen. So bleibt den BesucherInnen neben der Erinnerung an die Schönheit der Mathematik auch der Eindruck, etwas gelernt zu haben.

Flächen selbst gemacht

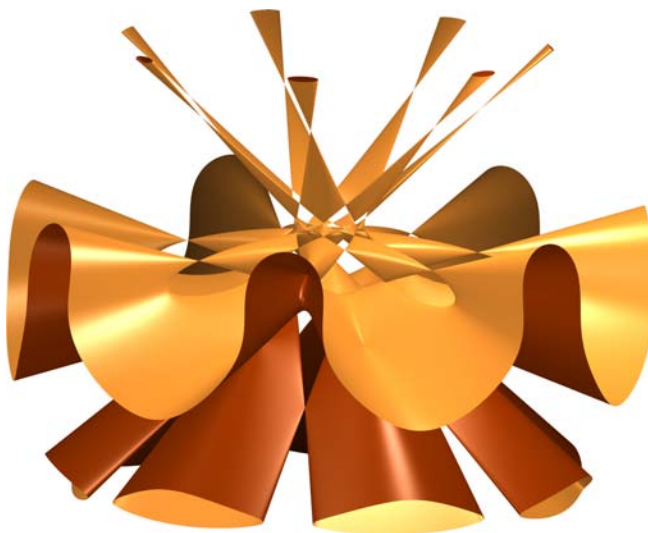
Interaktive Installationen ermöglichen es den BesucherInnen, selbst mathematisch-künstlerisch aktiv zu werden. Dazu wurde das Programm SURFER entworfen, mit dem man algebraische Flächen in Echtzeit berechnen, anzeigen und verändern kann. Auf einem großen Touch-Screen können die Besucher mit dem Finger die polynomialen Gleichungen eingeben oder abändern, Parameter verschieben, die Farben der Flächen bestimmen und die Figuren nach Belieben drehen. SURFER basiert auf dem Programm SURF von Stephan Endrass et al. und wurde von Henning Meyer (TU Kaiserslautern) und Christian Stussak (Oberwolfach, Halle) entwickelt. Das Besondere an SURF und SURFER sind die sehr stabilen und extrem schnellen Algorithmen zur Darstellung speziell der Singularitäten sowie die unübertroffen intuitive Eingabeoberfläche. Daneben bietet der SURFER verschiedene Galerien mit bekannten und neuen algebraischen Flächen, die auf einfache Weise verändert werden können.

Insbesondere die genial einfache Benutzerführung des SURFER hat die Redaktionen von ZEIT Online und Spektrum der Wissenschaften veranlasst, zusammen mit dem Mathematischen Forschungsinstitut Oberwolfach eigene Mathematik-Kunst-Wettbewerbe zu veranstalten.

Schulführungen

Für Schulklassen werden kostenlose Führungen angeboten. Nach einer kurzen Einführung mit Wiederholung zweidimensionaler Geometrie (Parabel, Kreisgleichung) mit dem Programm SURFER können die SchülerInnen selbst algebraische Flächen im dreidimensionalen Raum entwerfen. Kugeln werden dabei zu Ellipsoiden, die Gleichungen werden miteinander multipliziert und die singulären Punkte diskutiert. Durch kleine Parameteränderungen beim Doppelkegel wird

die „katastrophale“ Auswirkung im Bild veranschaulicht. Einblick in die aktuelle Forschung wird im Bereich der „Weltrekordflächen“ gegeben. Was ist die maximale Anzahl an singulären Punkten bei gegebenem Grad eines Polynoms? Oliver Labs (Universität des Saarlandes) erstellte dazu eine informative Galerie mit Erklärungen und vorgegebenen Parametern zum Verändern der Flächen. Natürlich ist dies nicht nur für SchülerInnen interessant.



*Die Labssche Septik mit 99 singulären Punkten
(derzeitiger Weltrekord).*

Am Ende der einstündigen Führung dürfen die SchülerInnen, genauso wie die anderen BesucherInnen, alle Programme ausprobieren und selbst geschaffene Bilder ausdrucken und mit nach Hause nehmen.

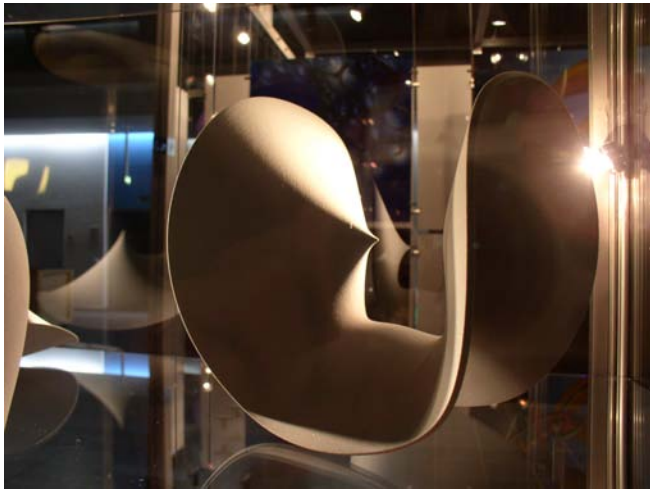


*Schülerinnen in Kassel
– ein Herz für die algebraische Geometrie!*

Flächen als 3D-Skulpturen

Die Firmen Voxeljet Technology in Augsburg und Alphaform in Feldkirchen befassen sich u. A. mit der generativen Fertigung von 3D-Modellen durch selektives Verkleben von Kunststoffpulver (PMMA) bzw. mittels Stereolithographie. Beide Firmen haben die Herausforderung angenommen, eine Auswahl algebraischer Flächen der Ausstellung in 3D zu drucken und als Skulpturen zu präsentieren. Die Schwierigkeit lag in der

Erstellung geeigneter Modelldatensätze. Für die Ausstellung wurden daher am Institut FORWISS der Universität Passau verschiedene Techniken umgesetzt, um algebraische Flächen in druckbare Daten zu wandeln. Zehn Skulpturen mit einem Durchmesser von ca. 25 cm werden exklusiv bei der Ausstellung gezeigt.



*3D-Skulpturen algebraischer Flächen,
hier das Bild Vis-à-Vis*

Posterset und Nachhaltigkeit

Auf der Webseite von IMAGINARY finden sich einführende Artikel der Mathematiker Herwig Hauser, Duco van Straten und Oliver Labs. Sie illustrieren einige der realen Hintergründe, geben einen freundlichen Einblick in die mathematische Werkstatt und einige aktuelle Forschungsprobleme. Diese Artikel wurden bereits mehr als 15.000 Mal von der Webseite heruntergeladen. Zusätzlich werden Tipps gesammelt, wie man das Programm SURFER in der Schule verwenden kann, Mathematik interaktiv und attraktiv vorzustellen. Ein Posterset mit 12 Motiven algebraischer Flächen wird für einen Unkostenbeitrag auf der Webseite <http://poster.imaginary2008.de> angeboten. Viele dieser Poster dekorieren bereits Schulklassen in ganz Deutschland, und einer eigenen Galerie der algebraischen Geometrie zu Hause steht nichts mehr im Wege.

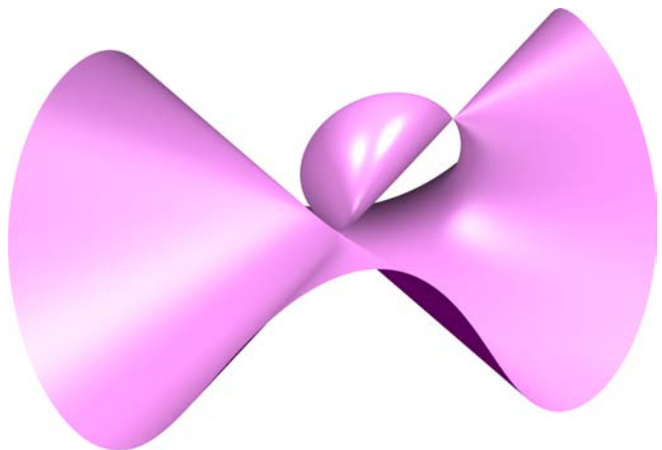
Erfahrungen mit IMAGINARY und SURFER

Die Ausstellung wurde bis jetzt in München, Berlin, Kaiserslautern, Stuttgart, Potsdam, Leipzig, Rust, Kassel und Köln gezeigt. Das Feedback dazu, gerade die interaktiven Elemente der Ausstellung betreffend, ist überwältigend und hat alle unsere Erwartungen übertroffen. Über 250 Schulklassen und mehr als 100.000 Personen besuchten die Ausstellung bisher. Die Möglichkeit, selbst algebraische Flächen zu erstellen, begeistert die BesucherInnen und schlägt sich in den zahlreichen Eintragungen in die Gästebücher nieder. Viele der BesucherInnen haben nach der Ausstellung bei Bilder-Wettbewerben mitgemacht. Die Mathematik-Kunst-Wettbewerbe mit Preisen bei Zeit Online und

Spektrum der Wissenschaft erhielten insgesamt über 2.000 Einsendungen, das Programm SURFER wurde bis jetzt über 30.000 Mal heruntergeladen. Diese Zahlen, aber auch die mehr als 15.000 Downloads der Erklärungsartikel bestätigen, dass ein großes Interesse an Mathematik besteht und dass Menschen sich gerne spielerisch und kreativ selbst mit mathematischen Inhalten befassen, wenn diese attraktiv und interaktiv dargestellt werden.

Die Wanderausstellung IMAGINARY wird vom Bundesministerium für Bildung und Forschung unterstützt.

Eine Kubik mit Parametern



*Eine Kubik mit zwei Singularitäten und der Formel
 $-x^3 - y^2 z - x z^2 + x z - x y a - 0,1 z^2 (b - 0,5) = 0$.
 Was passiert, wenn man die Parameter a und b
 zwischen 0 und 1 verändert?*

Weitere geplante Termine

Konstanz, 01.10.–19.10.2008
 München, 25.09.–21.10.2008
 Saarbrücken, 24.10.–16.11.2008
 Mainz, November–Dezember
 Passau, Dezember

Weblinks mit mehr Information

www.imaginary2008.de
<http://www.jahr-der-mathematik.de>

www.zeit.de/matheskulptur
www.spektrum.de/mathekunst

<http://surfer.imaginary2008.de>
<http://unterricht.imaginary2008.de>
<http://poster.imaginary2008.de>

www.freigeist.cc
www.algebraicsurface.net
www.mathematik.uni-kl.de/~greuel/de/projects.html

From the Lagrangian to Feynman Rules with FeynRules

Neil D. Christensen
Michigan State University

Claude Duhr
Université catholique de Louvain

neil@pa.msu.edu
claude.duhr@uclouvain.be



Introduction

Physics models in high-energy physics are in general based on *quantum field theory*. In this mathematical language, the model is described by a *Lagrangian*, which contains all the particles in the model and describes their mutual interactions. Each particle is represented by a *field*, defined over the four-dimensional space-time. The interactions among the particles are then represented by non-linear terms coupling different kinds of fields at the same space-time point. The simplest example of such a model is Quantum Electrodynamics (QED), which is the quantum theory describing the electromagnetic interaction among the electron, the positron (the antiparticle of the electron) and the photon (the ‘carrier’ of the electromagnetic interaction). The Lagrangian of QED is

$$\mathcal{L}_{\text{QED}} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{\psi}(i\gamma^\mu\partial_\mu - m)\psi - e\bar{\psi}\gamma^\mu\psi A_\mu, \quad (1)$$

where ψ , $\bar{\psi}$ and A_μ are the electron, positron and photon fields respectively, and $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$ is the electromagnetic field strength tensor. The first two terms, which are quadratic in the fields, describe the noninteractive dynamics of the electron, positron and photon, while the third term couples, in a trilinear way, these three fields and hence describes their interaction (e.g. annihilation of an electron-positron pair into a photon). A graphical representation of this interaction is given in Fig. 1 (a) and is called a vertex. Perturbative predictions of a model can be obtained by connecting together in all possible ways the vertices and propagators (lines connecting vertices) of a model using a set of rules called ‘Feynman rules’, where the vertices and propagators are derived from the Lagrangian. An example of such a Feynman diagram contributing to the scattering of an electron-positron pair in QED is shown in Fig. 1 (b).

Predictions for collider experiments often require the computation of thousands if not tens of thousands of Feynman diagrams. Such a monumental task is

especially suited to computers. There are several computer programs available that calculate Feynman diagrams automatically, both numerically and analytically. A few of these are CalcHEP/CompHEP, FeynArts/FormCalc, Herwig, MadGraph/MadEvent, Sherpa, and Omega/Whizard. These packages are responsible for generating the Feynman diagrams and making the predictions that will be compared with the results of experiment. However, only a very limited set of the possible physics models beyond the Standard Model is implemented in these tools, and hence their ability to determine which of all these models is correct is likewise limited. Implementing a new model into one of these codes requires to work out beforehand all the vertices that appear in the Lagrangian describing the model, and very often the input of one vertex at a time is necessary, making this a very tedious and error prone process for complicated models containing a large number of fields and interactions. On the other hand, deriving all the vertices and propagators from a Lagrangian is an algorithmic procedure very well suited to automation on a computer.

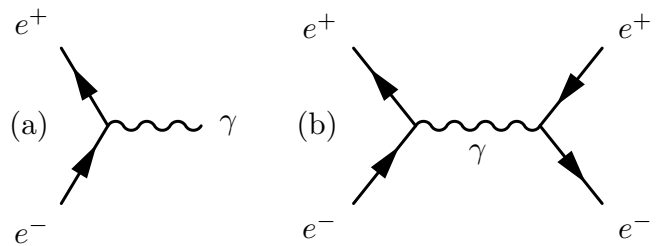


Fig. 1

- (a) The vertex describing the interaction of the electron (e^-), positron (e^+) and photon (γ).
(b) A Feynman diagram contributing to electron-positron scattering.

FeynRules is a new package based on *Mathematica* which takes a model file with the Lagrangian as input and derives the Feynman rules associated with it [1]. The package contains a set of functions allowing the user to test and build their model one piece at a time. In addition to these built-in functions, the user is invited

to exploit the full power of the underlying *Mathematica* code to extend these functions and to create his or her own new routines. In a second step, FeynRules can translate the model (with the derived vertices) into the model format corresponding to the Feynman diagram calculator chosen and allows in this way to implement the new model in any tool for which such a “translation interface” exists. In the following sections, we will briefly present the package, its features and the underlying algorithm.

The FeynRules model file

The basic input a user provides when implementing his or her model into FeynRules is the so called model file, a text file containing all the properties of the model (particles, parameters, *etc.*), and the Lagrangian written down using standard *Mathematica* language, augmented by some new symbols which are necessary when writing down a Lagrangian. The syntax and the structure of the model files, which is an extension of the original FeynArts model files, is based on the concept of classes. Each class consists of a physical quantity (*e.g.* a particle), together with the properties which define this quantity (*e.g.* the mass of the particle). There are three different kinds of classes which are used for a model implementation:

1. Particle classes: Each field appearing in the Lagrangian is assigned to a class, which specifies all the physical properties of the particle (*e.g.* spin, electric charge, mass, *...*). In order to avoid a proliferation of particle classes, particles having similar physical properties (but possibly different masses) can be assigned to the same class.
2. Parameter classes: A physical model is also defined by its parameters. Each parameter is assigned to a class and the properties of the class of parameters are specified such as the numerical value (*e.g.* the numerical value of the electromagnetic coupling constant e in the QED Lagrangian (1)) or the functional dependence of the parameter on other parameters (*e.g.* the fine structure constant is defined in terms of the electromagnetic coupling as $\alpha = e^2/4\pi$) *etc.*
3. Gauge group classes: Gauge symmetries play an important role in limiting the allowed interactions in a field theory and make possible a compact way of writing the Lagrangian. The user can specify each gauge group and its properties in this class.

At run time FeynRules reads in and processes each class defined in the model file and then saves this information in various *Mathematica* lists which are used later when deriving the Feynman rules and translating the model information to the format appropriate for one of the Feynman diagram calculators.

The second input the user needs to provide is the Lagrangian, written down in *Mathematica* language in a form close to the textbook expression. We will illustrate this procedure on the QED Lagrangian defined in Eq. (1). Let us start with the first term in \mathcal{L}_{QED} , the kinetic term for the photon field. In FeynRules, this term reads

$$-1/4 \text{FS}[A, \mu, \nu] \text{FS}[A, \mu, \nu]$$

The function `FS` appearing in this expression is predefined in FeynRules and returns the correct field strength tensor for the photon. The second term, the kinetic term for the electron field, is entered in FeynRules as

$$\text{I psi} \bar{\text{psi}}. \text{Ga}[\mu]. \text{del}[\text{psi}, \mu] - \text{m psi} \bar{\text{psi}}$$

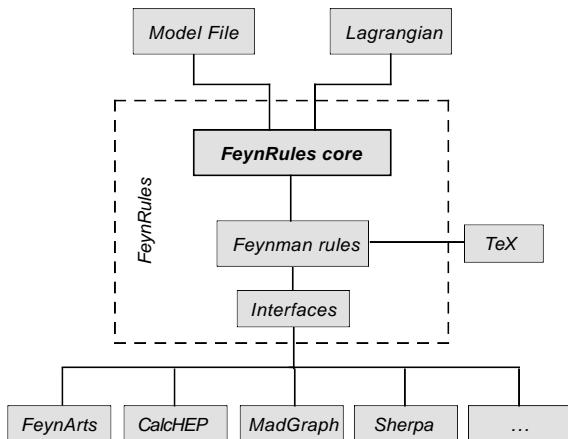
where `psi` represents the electron field and `m` its mass (those symbols have been defined by the user in the model file), and `Ga[μ]` and `del[., μ]` are special FeynRules commands representing the γ -matrices and the derivative ∂_μ . The dot-product appearing in this expression ensures that the anticommuting character of the electron field is correctly taken into account. Note that it is sufficient to define the electron field `psi` in the model file and FeynRules will automatically create the symbol `psibar` for the antiparticle. Finally, the interaction term can be entered in a similar way,

$$-e \text{psi} \bar{\text{psi}}. \text{Ga}[\mu]. \text{psi} A[\mu]$$

Putting those three terms together, we obtain the *Mathematica* expression for the QED Lagrangian, which looks formally the same as the textbook expression given in Eq. (1).

Derivation of the Feynman Rules

Once FeynRules reads the model file and the Lagrangian has been defined, it is ready to compute the Feynman rules, which is done by running the command `FeynmanRules[\mathcal{L} , options]`, where \mathcal{L} denotes the Lagrangian and *options* is a set of ‘selection rules’ that specify which subset of the full set of vertices should be computed (*e.g.* only those vertices involving a given number or combination of fields). The evaluation of the vertices from the Lagrangian involves two main steps outlined below.



In a first step, FeynRules prepares the Lagrangian. It does this by first checking that the Lagrangian fulfills the basic requirements of quantum field theory such as Lorentz and gauge invariance. It next applies the selection rules defined by the user. Applying the selection rules at this point allows to speed up the algorithm by avoiding the computation of undesired vertices. FeynRules then scans the remaining Lagrangian terms, identifying the fields in each, and groups terms with the same field content.

Once the Lagrangian is prepared, FeynRules applies standard quantum field theory rules in order to extract the interaction vertex. It multiplies on the right of each term by the creation operators for the fields, and then moves them all to the left of the expression by making recursive use of the canonical quantization commutation relations¹

$$[\psi_i(x), a_j^\dagger(p)] = \delta_{ij} u_i(p) e^{-ipx}, \quad (2)$$

where ψ_i denotes a field of type i , and $a_j^\dagger(p)$ is the creation operator associated to the field ψ_j . After having moved all creation operators to the left and dropping the external wave functions $u_i(p) e^{-ipx}$ for the fields, we are left with the interaction vertex. We note that this algorithm takes care of the (anti-)symmetrization required if identical fields are present in the vertex.

The vertices are then stored in a *Mathematica* list and can be directly used in further computations inside *Mathematica*, exported into a \LaTeX file or translated to the model file format appropriate for one of the Feynman diagram calculator as described in the next section.

Translation interfaces

Although the information about a model is generic, the way each Feynman diagram calculator stores that information is different and not compatible with one another. For this reason, FeynRules stores the information in a generic way and allows translation interfaces to be written. Currently translation interfaces for FeynArts/FormCalc, CalcHep/CompHep, MadGraph/MadEvent and Sherpa are available, but we hope to have many more in the future. Although the file formats for the different Feynman diagram calculators can be quite different, the translation interfaces have nevertheless a common structure which will be described in the following.

A translation interface begins by scanning the field and parameter classes, dropping information which is

not used in its Feynman diagram calculator and warning the user about possible incompatibilities. If the scan is successful, it stores the condensed information in new temporary lists and writes them to a file in the format appropriate for the particular Feynman diagram calculator. The translation interfaces also call the function `FeynmanRules` with the appropriate options for its Feynman diagram calculator and stores the resulting vertices in a list. It then scans and filters the list of vertices, keeping only the vertices that are supported in the Feynman diagram calculator being written to and warns the user about the vertices that are not supported. Finally, it writes the vertices to file in the appropriate format. The files are then ready for use in the Feynman diagram calculator².

Conclusion

In this article we have presented FeynRules, a *Mathematica* package which extracts Feynman rules from a Lagrangian, and allows to implement new physics models into several Feynman diagram calculators in an automated way. We have briefly reviewed the structure of the FeynRules input files, the algorithm used to derive Feynman rules from a Lagrangian, and how translation interfaces write this information in a format appropriate for one of the Feynman diagram calculators. Such interfaces are currently available for CalcHEP/CompHEP, FeynArts/FormCalc, MadGraph/MadEvent and Sherpa, but we hope that we can extend this list in the future. Several models have been implemented in FeynRules, translated to Feynman diagram calculators and successfully compared with the model implementations previously existing. The FeynRules code, the translation interfaces and a set of implemented models are available and can be downloaded from the FeynRules website [1].

Acknowledgments

C. Duhr is a research fellow of the “*Fonds National de la Recherche Scientifique*”, Belgium. N. D. Christensen was supported by the US National Science Foundation under grants PHY-0354226 and PHY-0555544.

Literatur

- [1] N. D. Christensen and C. Duhr, arXiv:0806.4194 [hep-ph], <http://feynrules.phys.ucl.ac.be/>

¹We give here the commutation relations for bosons. For fermions anticommutation relations are applied.

²In some cases, the output files have to be compiled.

Counting points on curves over finite fields

Frederik Vercauteren (Leuven)

frederik.vercauteren@esat.kuleuven.be

Introduction

This article presents a brief overview of efficient algorithms to count the number of points on a curve over a finite field developed over the last 25 years. These algorithms have diverse applications ranging from numerically verifying conjectures to applications in cryptography and coding theory. A very detailed exposition of (hyper)-elliptic curve cryptography, including an extensive bibliography, can be found in [1].

Applications in cryptography

In 1985, both Koblitz and Miller proposed to use the group of points on an elliptic curve over a finite field as the basis for a discrete logarithm based cryptosystem. Let \mathbb{F}_q be a finite field of characteristic $p > 3$, then an elliptic curve E over \mathbb{F}_q can be given by an equation of the form

$$E : y^2 = x^3 + ax + b$$

for constants $a, b \in \mathbb{F}_q$ with $4a^3 + 27b^2 \neq 0$. A slightly different equation holds for $p = 2$ and $p = 3$. All points $(x, y) \in \mathbb{F}_q^2$ that satisfy the above equation together with a special point \mathcal{O} , the so called point at infinity, constitute the \mathbb{F}_q -rational points on E , denoted $E(\mathbb{F}_q)$. The set $E(\mathbb{F}_q)$ can be endowed with a group law by using the chord-and-tangent procedure: to add two points P and Q on the curve, one constructs the line through P and Q (or the tangent line if $P = Q$), finds the third intersection point with the curve and reflects this around the x -axis. The resulting point R is then the sum of P and Q . The point \mathcal{O} acts as the neutral element and the negative of a point is obtained by reflecting the point around the x -axis. Repeated addition/subtraction results in scalar multiplication, i.e. $[m]P$ with $m \in \mathbb{Z}$.

The inverse operation, i.e. to recover m from $Q = [m]P$ and P , is called the elliptic curve discrete logarithm problem (ECDLP). The ECDLP and related problems such as the Diffie-Hellman problem (CDH), i.e. given points $[a]P$, $[b]P$, determine $[ab]P$, lie at the heart of several cryptographic primitives, such as the Diffie-Hellman key exchange protocol.

The hardness of the ECDLP (and also the CDH) is mainly determined by the group order $|E(\mathbb{F}_q)|$. If

$$|E(\mathbb{F}_q)| = \prod_i p_i^{e_i},$$

then one can project the ECDLP in subgroups of order p_i . Using a generic method like Pollard Rho, each of these subproblems can be solved in time $O(p_i^{1/2})$. The Chinese remainder theorem finally allows to recover the full

discrete log in $E(\mathbb{F}_q)$. In practice, one therefore chooses elliptic curves such that $|E(\mathbb{F}_q)| = c \cdot r$ with r a large prime (having between 256 and 512 bits) and a small cofactor c . This also shows that naive algorithms to compute $|E(\mathbb{F}_q)|$, such as enumeration, are totally inadequate to generate elliptic curves for use in cryptography.

More generally, one can also use the group of points on the Jacobian $J_{\overline{C}}$ of a hyperelliptic curve \overline{C} as proposed by Koblitz. Again it is necessary that $|J_{\overline{C}}(\mathbb{F}_q)|$ is divisible by a large prime to assure the hardness of the DLP.

Zeta Function and Weil Conjectures

Let \overline{C} be a smooth projective curve of genus g over a finite field \mathbb{F}_q . For each extension field \mathbb{F}_{q^k} we denote by $N_k = |\overline{C}(\mathbb{F}_{q^k})|$ the number of \mathbb{F}_{q^k} -rational points on \overline{C} . The generating series

$$Z(\overline{C}/\mathbb{F}_q; t) := \exp \left(\sum_{k \geq 1} N_k \frac{t^k}{k} \right) \in 1 + t\mathbb{Q}[[t]]$$

is called the zeta function of \overline{C} . A fortiori, the zeta function should be interpreted as a formal power series with coefficients in \mathbb{Q} . In 1949, Weil formulated a list of conjectures for the zeta function of a smooth projective variety over a finite field and proved these conjectures for curves and abelian varieties. Weil's conjectures (for curves) can be summarized as follows: $Z(\overline{C}/\mathbb{F}_q; t)$ is a rational function of the form

$$Z(\overline{C}/\mathbb{F}_q; t) = \frac{L(t)}{(1-t)(1-qt)},$$

with

$$L(t) = c_0 + c_1 t + \dots + c_{2g} t^{2g} \in \mathbb{Z}[t]$$

and $c_0 = 1$. The coefficients of L satisfy $c_{2g-i} = q^{g-i} c_i$ for $i = 0, \dots, g$ and if

$$L(t) = \prod_{i=1}^{2g} (1 - \alpha_i t),$$

then $|\alpha_i| = \sqrt{q}$. Furthermore, the α_i can be labeled such that $\alpha_i \alpha_{i+g} = q$ for $i = 0, \dots, g$.

The zeta function $Z(\overline{C}/\mathbb{F}_q; t)$ contains important geometric information about \overline{C} , but also its Jacobian $J_{\overline{C}}$. From a practical point of view, the property $L(1) = |J_{\overline{C}}(\mathbb{F}_q)|$ is most important.

The Weil conjectures have some interesting algorithmic implications: it suffices to compute c_i for $i = 1, \dots, g$, each of which is an integer bounded by

$$|c_i| \leq \binom{2g}{i} q^{i/2} \leq 2^{2g} q^{g/2}.$$

This shows that the size of L is $O(g \log q)$, so the best possible algorithm to compute the zeta function will be polynomial both in g and $\log q$.

Weil Cohomologies

Apart from stating his now proven conjectures, Weil also formulated a theoretical framework to prove them, namely the notion of a Weil cohomology theory. The central idea is as follows: consider the Frobenius automorphism

$$\phi_q : \overline{\mathbb{F}}_q \rightarrow \overline{\mathbb{F}}_q, \quad x \mapsto x^q$$

and extend ϕ_q to the Frobenius morphism on $\overline{C}/\mathbb{F}_q$. Since \mathbb{F}_q is the fixed field of ϕ_q , the \mathbb{F}_q -rational points on $\overline{C}/\mathbb{F}_q$ are precisely the fixed points of ϕ_q . For a compact complex analytic manifold \mathcal{M} , there exists a general formula for the number of fixed points of an analytic map f , called the Lefschetz Fixed Point Theorem. Lefschetz' formula expresses the number of fixed points of f as an alternating sum of traces of the induced linear map f^* on the de Rham cohomology groups $H_{DR}^i(\mathcal{M})$. Weil envisaged to mimic this situation for varieties over finite fields, i.e. to construct a good cohomology theory (over a characteristic zero field) such that the number of fixed points of the Frobenius morphism is given by a Lefschetz fixed point formula. Currently there exist several good Weil cohomologies, some of which lead to efficient algorithms to compute the zeta function of a curve over a finite field.

ℓ -adic Approach

Let ℓ denote a prime different from p and let \mathbb{Q}_ℓ be the field of ℓ -adic numbers. Grothendieck introduced the ℓ -adic cohomology groups $H^i(\overline{C}, \mathbb{Q}_\ell)$ and proved that the number of \mathbb{F}_{q^k} -rational points N_k is given by a Lefschetz trace formula

$$N_k = \sum_{i=0}^2 (-1)^i \text{Tr}(\phi_q^{k*}; H^i(\overline{C}, \mathbb{Q}_\ell)).$$

It can be shown that ϕ_q^k acts as the identity on $H^0(\overline{C}, \mathbb{Q}_\ell)$ and as multiplication by q^k on $H^2(\overline{C}, \mathbb{Q}_\ell)$. Substituting this formula in the definition of the zeta function then shows that

$$L(t) = \det(1 - \phi_q^* t; H^1(\overline{C}, \mathbb{Q}_\ell)).$$

Although the above gives a closed expression for $L(t)$, it does not directly lead to an algorithm since the description of $H^1(\overline{C}, \mathbb{Q}_\ell)$ is very abstract and not amenable to

computation. However, for a smooth projective curve \overline{C} we have the following isomorphism

$$H^1(\overline{C}, \mathbb{Z}_\ell) \simeq T_\ell(J_{\overline{C}}),$$

where $T_\ell(J_{\overline{C}})$ denotes the ℓ -adic Tate module of $J_{\overline{C}}$, which by definition is the inverse limit of the projective system of groups $J_{\overline{C}}[\ell^k]$, i.e. the ℓ^k -torsion on $J_{\overline{C}}$. For $\ell \neq p$, the structure of $J_{\overline{C}}[\ell^k]$ is given by $(\mathbb{Z}/\ell^k \mathbb{Z})^{2g}$, so $T_\ell(J_{\overline{C}})$ as a \mathbb{Z}_ℓ -module is isomorphic to \mathbb{Z}_ℓ^{2g} . Furthermore, we obtain a representation

$$T_\ell : \text{End}_{\mathbb{F}_q}(J_{\overline{C}}) \rightarrow \text{End}_{\mathbb{Z}_\ell}(T_\ell(J_{\overline{C}})),$$

so $L(t)$ can be computed explicitly as

$$L(t) = \det(1 - T_\ell(\phi_q)t),$$

independently of ℓ . Exploiting this expression for fixed ℓ does not lead to an efficient algorithm since one would need to approximate $T_\ell(J_{\overline{C}})$ by $J_{\overline{C}}[\ell^k]$ with k such that $\ell^k > 2 \cdot 2^{2g} \cdot q^{g/2}$ to be able to recover $L(t)$ uniquely. The torsion group $J_{\overline{C}}[\ell^k]$ is isomorphic to $(\mathbb{Z}/\ell^k \mathbb{Z})^{2g}$, so it requires at least $O(2^{g^2} q^{g^2} \log q)$ space to describe, which clearly is infeasible.

The solution to this problem is to choose many different small primes ℓ_i and compute $L(t) \bmod \ell_i$ as the characteristic polynomial of ϕ_q on $J_{\overline{C}}[\ell_i]$. Finally, one recovers $L(t)$ uniquely using the Chinese remainder theorem if

$$\prod_i \ell_i > 2 \cdot 2^{2g} \cdot q^{g/2}.$$

The prime number theorem implies that the size of the largest prime ℓ_i is of the order $O(g \log q)$.

This approach was first proposed by Schoof for elliptic curves resulting in an algorithm to compute the cardinality of $E(\mathbb{F}_q)$ in time $\tilde{O}(\log^5 q)$ assuming fast arithmetic. Although polynomial time in $\log q$, Schoof's algorithm can only handle q up to 256 bits in reasonable time (several hours), and is not fast enough to cover the whole cryptographic spectrum where q can have up to 512 bits. The main problem is again the size of $E[\ell_i]$ which contains ℓ_i^2 points. Schoof's algorithm was improved both by Atkin and Elkies by considering the action of Frobenius on a smaller object than $E[\ell_i]$. Atkin used the projective line

$$(E[\ell_i] \setminus \{\mathcal{O}\})/\mathbb{F}_{\ell_i}^*,$$

whereas Elkies considered eigenspaces of Frobenius on $E[\ell_i]$, which exist for roughly half the primes ℓ_i and consist of ℓ_i points only. Combining both improvements leads to the Schoof-Elkies-Atkin (SEA) algorithm with a complexity of $\tilde{O}(\log^4 q)$. The SEA-algorithm is efficient enough to easily find secure elliptic curves in the whole cryptographic range.

The ℓ -adic approach has also been worked out, at least in theory, for abelian varieties by Pila and by Adleman and Huang for Jacobians of curves, resulting in an algorithm with complexity $(\log q)^{O(g^2 \log q)}$. Unfortunately, this algorithm is only just practical enough to generate secure genus 2 hyperelliptic curves as shown by Gaudry and Schost.

p-adic Approach

Currently there are three different types of efficient p -adic algorithms depending on how the curve \bar{C} and the Frobenius endomorphism ϕ_q are lifted to an unramified extension \mathbb{Q}_q of the p -adic field \mathbb{Q}_p of degree n where $q = p^n$. Denote with \mathbb{Z}_q the valuation ring of \mathbb{Q}_q , then $\mathbb{Z}_q/(p) \simeq \mathbb{F}_q$. A common feature of these algorithms is that their complexity depends exponentially on p (and are thus only efficient for small p), but is polynomial in n . The underlying reason for this behavior is that the q -th power Frobenius endomorphism ϕ_q can be decomposed in n copies of the p -th power Frobenius morphism ϕ_p

$$\bar{C} \xrightarrow{\phi_p} \bar{C}^p \xrightarrow{\phi_p} \bar{C}^{p^2} \dots \bar{C}^{p^{n-1}} \xrightarrow{\phi_p} \bar{C},$$

where \bar{C}^{p^k} denotes the curve obtained from \bar{C} by raising its coefficients to the p^k -th power. As such it suffices to study the action of one p -th power Frobenius morphism to recover the action (typically by a norm argument) of the q -th power Frobenius.

The first type of p -adic algorithm is based on the theory of the **canonical lift** and was introduced by Satoh for elliptic curves over finite fields with $p > 3$. Let E be an elliptic curve over \mathbb{F}_q , then one can consider many arbitrary lifts \mathcal{E} to \mathbb{Q}_q , i.e. any curve \mathcal{E} with reduction modulo p equal to E is a valid lift. Since \mathbb{Q}_q is a field of characteristic 0, the general situation is that $\text{End}(\mathcal{E})$ only contains the multiplication by m endomorphisms, i.e. $\text{End}(\mathcal{E}) = \mathbb{Z}$. However, if E is ordinary ($E[p] \neq \{\mathcal{O}\}$), then Deuring showed there exists a unique lift \mathcal{E} to \mathbb{Q}_q such that $\text{End}(\mathcal{E})$ is isomorphic to $\text{End}(E)$ via reduction modulo p . This unique lift is called the canonical lift of E . Denote with \mathcal{F}_q the lift of ϕ_q to $\text{End}(\mathcal{E})$, then the characteristic polynomial of \mathcal{F}_q equals the characteristic polynomial of ϕ_q :

$$\mathcal{F}_q \circ \mathcal{F}_q - [t_E] \circ \mathcal{F}_q + [q] = 0.$$

Here, t_E is the trace of Frobenius which satisfies

$$E(\mathbb{F}_q) = q + 1 - t_E.$$

To compute t_E , Satoh analyzes the action of \mathcal{F}_q on an invariant differential ω on \mathcal{E} . Since up to constants there is only one such invariant differential, there exists a $c \in \mathbb{Q}_q$ with $\mathcal{F}_q^*(\omega) = c\omega$. Using the above characteristic equation, one then concludes that

$$t_E = c + q/c,$$

so it suffices to compute c . Furthermore, by using the decomposition of ϕ_q one can show that $c = \text{Norm}_{\mathbb{Q}_q/\mathbb{Q}_p}(c_0)$ with c_0 the action of the canonical lift of the p -th power Frobenius ϕ_p on the invariant differential. For fixed p , Satoh's original algorithm runs in time $\tilde{O}(n^3)$. Improvements and extensions to characteristic 2 and 3 were proposed by many people (see [1] for an overview) and finally resulted in Harley's algorithm with complexity $\tilde{O}(n^2)$. Mestre described a related algorithm based on a 2-adic version of the AGM

(Arithmetic-Geometric Mean) and showed how it generalized to ordinary hyperelliptic curves over finite fields of characteristic 2. An optimized version of this algorithm runs in time $\tilde{O}(2^9 n^2)$ and is only efficient for small g .

The second type of p -adic algorithm is based on **p -adic cohomology**. Kedlaya described an algorithm based on Monsky-Washnitzer cohomology to compute the zeta function of a hyperelliptic curve and Lauder and Wan used Dwork-Reich cohomology to devise an algorithm for the class of Artin-Schreier curves. Since Kedlaya's approach has turned out most effective, we will only describe Monsky-Washnitzer cohomology for a non-singular plane affine curve defined by $\bar{C}(x, y) = 0$. Monsky and Washnitzer construct a cohomology group $H_{MW}^1(\bar{C}/\mathbb{Q}_q)$ together with a lift \mathcal{F}_q of Frobenius such that the zeta function of the *affine* curve \bar{C} is given by

$$Z(\bar{C}/\mathbb{F}_q; t) = \frac{\det(1 - q\mathcal{F}_q^{*-1}t; H_{MW}^1(\bar{C}/\mathbb{Q}_q))}{1 - qt}.$$

The group $H_{MW}^1(\bar{C}/\mathbb{Q}_q)$ is constructed as follows: take an arbitrary lift \mathcal{C} to \mathbb{Q}_q and consider the coordinate ring

$$A = \mathbb{Q}_q[x, y]/(\mathcal{C}(x, y))$$

of \mathcal{C} . Define

$$A^\dagger = \frac{\mathbb{Z}_q\langle x, y \rangle^\dagger}{(\mathcal{C}(x, y))},$$

where $\mathbb{Z}_q\langle x, y \rangle^\dagger$ is the *weak completion* of $\mathbb{Z}_q[x, y]$. It consists of overconvergent power series

$$\sum a_{i,j} x^i y^j \in \mathbb{Z}_q[[x, y]]$$

where $\text{ord}_p(a_{i,j})$ grows linearly with $i + j$. The idea behind this convergence condition is that $\mathbb{Z}_q\langle x, y \rangle^\dagger$ should be closed under integration. Let $D^1(A^\dagger)$ be the universal module of differentials on A^\dagger over \mathbb{Z}_q and let $d : A^\dagger \rightarrow D^1(A^\dagger)$ be the usual exterior derivation. Then $H_{MW}^1(\bar{C}/\mathbb{Q}_q)$ is defined as $\frac{D^1(A^\dagger)}{d(A^\dagger)} \otimes_{\mathbb{Z}_q} \mathbb{Q}_q$. The first stage in Kedlaya's algorithm is to compute a basis B for $H_{MW}^1(\bar{C}/\mathbb{Q}_q)$ and reduction formulae to express any differential form on this basis B . In the second stage, a sufficiently precise approximation of $\mathcal{F}_q(x)$ and $\mathcal{F}_q(y)$ is computed, together with a matrix M over \mathbb{Q}_q such that $\mathcal{F}_q^*(B) = MB$. By computing the characteristic polynomial of M one then recovers $Z(\bar{C}/\mathbb{F}_q; t)$. Again it suffices to study the action of a lift \mathcal{F}_p of the p -th power Frobenius instead of \mathcal{F}_q and to recover M by a norm argument. For fixed p , Kedlaya's algorithm runs in time $\tilde{O}(g^4 n^3)$. Many extensions to wider classes of curves, such as non-degenerate curves, exist. Furthermore, the bad dependency on p , namely $\tilde{O}(p)$, can be improved to $\tilde{O}(p^{1/2})$ as shown by Harvey.

The third type of p -adic algorithm computes the zeta function of a one-parameter family of curves by **deformation** of a single fiber in the family. This approach was first proposed by Lauder and worked out in detail for hyperelliptic curves by Hubrechts. Let $\bar{C}(x, y, t) \in \mathbb{F}_q[t][x, y]$ define a family of smooth affine curves over

some open dense subset of the affine t -line. Instead of studying the fibers separately as in the second method, the aim of the deformation is to describe how the action of Frobenius on $H_{MW}^1(\overline{C}(x, y, t)/\mathbb{Q}_q)$ alters as t varies. The crucial point is that the matrices of Frobenius on $H_{MW}^1(\overline{C}(x, y, t)/\mathbb{Q}_q)$ satisfy a differential equation, which expresses the quasi-commutativity of the so called Gauss-Manin connection with the Frobenius action. The resulting algorithm thus proceeds in two stages: first compute the matrix of Frobenius of some fiber, typically $t = 0$ by a type two method and then solve the differential equation to obtain the matrix of Frobenius of the fiber one is interested in. The complexity of this algorithm for a family defined over \mathbb{F}_{p^a} and fibers over $\mathbb{F}_{p^{an}}$ is $\tilde{O}(n^{2.667} g^{6.376} a^3)$. The main use of this algorithm is to search for curves with $|J_C(\mathbb{F}_q)|$ divisible by a large prime by choosing a family $\overline{C}(x, y, t)$ over \mathbb{F}_p and fibers over \mathbb{F}_{p^n} . After a precomputation that depends only on the family, computing the zeta function of each new fiber is very efficient.

Algorithms in Computer Algebra Systems

The most complete computer algebra system with respect to point counting algorithms is Magma: it includes all algorithms including most optimizations described in this article. PARI/GP contains an implementation of Schoof's algorithm and freely available code for Satoh's algorithm exists. SAGE contains a native implementation of Kedlaya's algorithm, including Harvey's optimization.

Literatur

- [1] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren (Eds.) *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

Neues über Systeme

Galoisgruppen in Magma

Claus Fieker
University of Sydney

Jürgen Klüners
Heinrich-Heine-Universität Düsseldorf

claus@maths.usyd.edu.au
klueners@math.uni-duesseldorf.de



Die Galoistheorie ist eines der klassischen Gebiete der Mathematik. So konnte vor mehreren hundert Jahren gezeigt werden, dass alle Polynome bis zum Grad 4 durch Radikale (Wurzelausdrücke) gelöst werden können. Durch die Galoistheorie wird jeder Gleichung eine Permutationsgruppe, die sogenannte Galoisgruppe, zugeordnet. Eine Gleichung ist genau dann durch Radikale lösbar, wenn ihre Galoisgruppe auflösbar ist. In diesem Bericht wollen wir einen Überblick über das Berechnen von Galoisgruppen in Magma (<http://magma.maths.usyd.edu.au/magma/>) geben.

Algorithmisch ist die Berechnung ein interessantes Problem. Jeder Zuhörer einer Algebra-Vorlesung hat sicherlich schon mühsam per Hand die Galoisgruppe eines kleinen Polynoms berechnet bzw. gezielt erraten. Obwohl die ursprünglichen Beweise „fast“ konstruktiv sind, sind sie selbst auf einem modernen Computer nicht praktikabel. Pakete zur Berechnung von Galoisgruppen

sind daher schon seit vielen Jahren in Computeralgebrasystemen wie Maple, Magma, Kash oder Gap enthalten. Allerdings gab es immer Gradschranken, die im Laufe der Jahre von Grad 7 oder 8 je nach Paket bis auf Grad 23 angewachsen sind.

Ab der Version 2.13 von Magma ist es nun grad-unabhängig möglich, Galoisgruppen von Polynomen über den rationalen Zahlen zu bestimmen, wobei es egal ist, ob das Polynom irreduzibel ist oder nicht. In der aktuellen Magma-Version (2.14) sind diese Algorithmen auch für Polynome über Zahlkörpern oder über dem rationalen Funktionenkörper $\mathbb{Q}(t)$ anwendbar.

Wir verwenden eine Variante des klassischen Ansatzes von Richard Stauduhar mit der wesentlichen Neuerung, dass alle benötigten Daten (Gruppentheorie, Invariantentheorie) zur Laufzeit berechnet werden und nicht wie sonst üblich Tabellen entnommen werden. Der Befehl *GaloisGroup* berechnet in allen Fällen

die Galoisgruppe als explizite Gruppe von Permutationen der Nullstellen des Polynomes in einem geeigneten p -adischen Körper (Potenzreihenkörper über einem p -adischen Körper für $\mathbb{Q}(t)$). Die explizite Operation auf den Nullstellen kann dann auch für weitere Berechnungen genutzt werden. So können wir zum Beispiel entscheiden, ob ein polynomieller Ausdruck in den Nullstellen verschwindet oder nicht. Weiterhin enthält Magma die Funktion *GaloisSubgroup*, welche einen Fixkörper, d. h. eine Gleichung für einen beliebigen Teilkörper des Zerfällungskörpers berechnet.

Mit Hilfe der Funktion *GaloisQuotient* können wir Polynome konstruieren, die zu anderen Permutationsdarstellungen derselben Gruppe korrespondieren: Die alternierende Gruppe auf 5 Punkten, $A(5)$, kann auch als Untergruppe der symmetrischen Gruppe auf 6 Punkten realisiert werden. Wir können nun ausgehend von einem Polynom vom Grad 5 mit Galoisgruppe $A(5)$ ein Polynom vom Grad 6 mit isomorpher Galoisgruppe konstruieren. Als letzte Anwendung wollen wir noch die Auflösbarkeit durch Radikale erwähnen, welche im auflösbaren Fall mit Hilfe der Funktion *SolveByRadicals* berechnet werden kann.

Um weitere Verbesserungen des Verfahrens bzw. Anwendungen zu ermöglichen, sind viele der für die Berechnung von Galoisgruppen notwendigen Teilschritte separat für Benutzer zugreifbar. So sind zum Beispiel alle Routinen, die invariante Polynome für (maximale) Untergruppen finden, verfügbar. Darauf aufbauend kann

dann ein Stauduhar-Schritt für ein beliebiges Gruppenpaar aufgerufen werden.

Wir merken an, dass der erstmalig implementierte Algorithmus für reduzible Polynome dafür verwendet werden kann, um zu testen, ob die Zerfällungskörper zweier Polynome linear disjunkt sind.

Mit der vorgestellten Implementierung wurden Galoisgruppen von Polynomen vom Grad > 120 erfolgreich berechnet. Je nach dem Grad variiert die Laufzeit von wenigen Sekunden (Grad < 10) bis zu mehreren Stunden (Grad > 100). Die tatsächliche Laufzeit ist aber nicht nur vom Grad, sondern auch von den zu berechnenden Galoisgruppen abhängig und kann daher selbst bei fixiertem Grad stark variieren.

Wir illustrieren einige der Funktionen an einem Beispiel. Ausgangspunkt ist das Polynom $x^4 + 3x^3 - 4x^2 - 11x + 4$, welches die alternierende Gruppe A_4 als Galoisgruppe hat. Wir berechnen zunächst die Galoisgruppe und dann den Fixkörper des Zerfällungskörpers (ohne diesen zu berechnen) unter einer Untergruppe der Ordnung 2. Dieser wird von einem Polynom vom Grad 6 erzeugt, welches wir dann durch Radikale auflösen. Die Magma-Ausgaben sind an verschiedenen Stellen gekürzt. Wir merken an, dass die Nullstellen von f in der unverzweigten Erweiterung vom Grad 3 über den 7-adischen Zahlen approximiert werden. Wir sehen, dass die Nullstellen des Polynoms g als Polynome in a_1, \dots, a_4 dargestellt sind, wobei die a_i selber Radikalgleichungen erfüllen.

```
> Zx<x> := PolynomialRing(Integers());
> f := x^4 + 3*x^3 - 4*x^2 - 11*x + 4;
> G, R := GaloisGroup(f);
> G;
Permutation group G acting on a set of cardinality 4
Order = 12 = 2^2 * 3
  (1, 2) (3, 4)
  (1, 2, 3)
> TransitiveGroupDescription(G);
A(4)
> R;
[ -2883 + O(7^5), 840*$.1^2 - 6136*$.1 - 4585 + O(7^5), ... ]
> Universe(R);
Unramified extension defined by the polynomial
  (1 + O(7^20))*x^3 ... over 7-adic ring
> s := Subgroups(G:OrderEqual := 2);
> g := GaloisSubgroup(f, s[1]'subgroup);
> g;
x^6 - 17*x^5 - 3*x^4 + 1137*x^3 - 2726*x^2 - 18984*x + 53512
> TransitiveGroupDescription(GaloisGroup(g));
A_4(6) = [2^2]3
> K, R := SolveByRadicals(g:Name := "a");
> K:Maximal;
...
K : a1^2 + 1/3912*(3*a4 - 25)*a3^2 + 1/6*a3 - 59/12
$1 : a2^2 + 1/1956*(-7*a4 + 4)*a3^2 + 1/12*(-a4 - 1)*a3 - 59/12
$2 : a3^3 + 1956*a4 + 16300
%$3 : a4^2 + 3
> R;
[ a1 + 1/1956*(-7*a4 + 4)*a3^2 + 1/12*(-a4 - 1)*a3 + 17/6, ... ]
> Evaluate(g, R[1]);
0
```

Vorstellung des GAP-Projektes

Frank Lübeck (Aachen)

frank.luebeck@rwth-aachen.de

GAP ist ein Computeralgebrasystem, dessen Entwicklung vor etwa 20 Jahren am Lehrstuhl D für Mathematik, RWTH Aachen, begonnen wurde, und zunächst zum Rechnen in endlichen Gruppen gedacht war. Heute ist die Wartung und Weiterentwicklung von GAP ein internationales Projekt, und das System deckt einen weiten Bereich mathematischer Anwendungen ab. In diesem Artikel möchte ich einige Aspekte des Aufbaus und der Funktionsweise des Systems sowie der Organisation des Projektes vorstellen.

GAP ist freie Software, die unter den Bedingungen der [GPL]-Lizenz kostenlos und mit vollständigem Quellcode über die Webseite [GAP] erhältlich ist. Auf dieser Webseite sind auch viele weitere Informationen zu finden.

Aufbau des Systems

Ich möchte hier vier Bestandteile von GAP erwähnen, kann in diesem kurzen Artikel aber jeweils nur wenige Beispiele dafür geben, was in diesen Teilen implementiert ist.

Kernel. GAP besteht zunächst aus einem Kern, der in C implementiert ist. Dieser enthält eine automatische Speicherverwaltung, stellt grundlegende Objekte wie zum Beispiel Listen, Records, rationale Zahlen, Permutationen und Funktionen für diese zur Verfügung, und er enthält einen Interpreter für die GAP-Programmiersprache (ähnlich Pascal oder Maple).

Library. Der größte Teil der Funktionalität des Systems ist in der GAP-Programmiersprache implementiert, zur Laufzeit wird dieser Code vom GAP-Kern interpretiert. Zum Beispiel enthält diese GAP-Bibliothek Funktionen zum Erzeugen und Rechnen mit Gruppen (gegeben durch erzeugende Permutationen, Matrizen, Präsentationen, ...), endlichen Körpern, Einheitswurzel-erweiterungen von \mathbb{Q} , Charaktertafeln endlicher Gruppen, Homomorphismen und vieles mehr.

Der Ansatz, Funktionalität überwiegend in der GAP-Sprache zu implementieren, hat den Vorteil, dass der Code oft recht gut lesbar und entsprechend leicht weiterverwendbar, veränderbar und erweiterbar ist. Die Kosten durch die Interpretation halten sich in Grenzen.

Data. Neben den implementierten Algorithmen sind verschiedene Datensammlungen ein weiteres wichtiges Standbein von GAP. Zum Beispiel können alle Gruppen mit gewissen Eigenschaften abgerufen werden (etwa mit Ordnung n ($n \leq 2000$) [SGP], oder transitiv auf n Punkten ($n < 32$) [TGP], ...), oder es kann auf eine große Zahl von Charaktertafeln endlicher Gruppen [ATL], [MAT] zugegriffen werden.

Packages. Die Funktionalität von GAP wird noch erheblich erweitert durch GAP-Pakete. Diese sind nicht

Teil der eigentlichen GAP-Distribution, sondern werden unabhängig von GAP-Nutzern entwickelt und können anderen Nutzern zu Verfügung gestellt werden.

Ein GAP-Paket kann mit einem einfachen Kommando in eine laufende GAP-Session geladen werden, und der Code, inklusive der Dokumentation, steht damit genau so zur Verfügung wie die Funktionen aus der GAP-Bibliothek.

Über die GAP-Webseite werden zur Zeit die jeweils aktuellen Versionen von etwa 75 Paketen, die von über 100 Autoren entwickelt wurden, zur Verfügung gestellt. Einige davon erweitern GAP auf neue mathematische Gebiete, zum Beispiel Graphen- oder Codierungstheorie, andere verbessern Funktionalität aus der GAP-Bibliothek, zum Beispiel für spezielle Typen von Gruppen, wieder andere stellen einzelne besonders effiziente Algorithmen zur Verfügung, und es gibt auch Pakete, die Schnittstellen zwischen GAP und anderer Software realisieren.

Bei der Weiterentwicklung des Kerns und der Bibliothek von GAP liegt der Schwerpunkt darauf, grundlegende Funktionalität zu verbessern. Ganz neue Funktionalität wird jedoch meist durch Pakete realisiert (viele GAP-Entwickler sind gleichzeitig Paket-Autoren). Dies hat viele Vorteile: die Zuständigkeit für Code ist klar, als Paket-Autor bekommt man bessere Anerkennung für die Entwicklung des Codes, Schnittstellen zum Code sind oft klarer dokumentiert, das Laden eines Paketes ist optional, Code ist leichter austauschbar.

In einem Paket können ganz neue mathematische Objekte und Funktionen dafür implementiert werden, ohne dass eine Hilfe der GAP-Entwickler benötigt wird oder eine Änderung im Rest von GAP notwendig ist. Andererseits wird den Paket-Autoren natürlich geholfen, wenn sie Fragen haben oder Vorschläge für Verbesserungen des Kerns oder der Bibliothek. (Es gibt sogar einige Pakete, die den GAP-Kern erweitern.)

Operationen und Methoden

In diesem Abschnitt möchte ich eine Besonderheit vorstellen, die ich in keinem anderen Computeralgebrasystem gesehen habe, die *Methoden-Selektion* in GAP.

Alle Objekte in GAP (ganze Zahlen, Permutationen, Listen, Gruppen, Körper, Homomorphismen, ...) haben einen *Typ*, durch den unterschieden werden soll, was man mit diesen Objekten machen kann. Den Typ können wir uns vereinfacht als eine Bit-Liste vorstellen, in der jede Position eine gewisse Interpretation hat. Zum Beispiel gibt es Bits, die gesetzt sind, wenn ein Objekt eine Gruppe ist, oder wenn es eine Permutation in einer

gewissen internen Datenstruktur ist, oder wenn es eine Gruppe ist, von der bekannt ist, dass sie auflösbar ist, ...

Es gibt *Funktionen* in GAP, bei denen die Argumente gewisse Bedingungen erfüllen müssen, damit etwas Sinnvolles berechnet wird, zum Beispiel `Factorial`, das mit einer nicht-negativen ganzen Zahl aufgerufen werden muss, sonst gibt es eine Fehlermeldung. Der Benutzer ist verantwortlich, sinnvolle Argumente zu übergeben.

Weiter gibt es aber auch *Operationen*. Diese sind Platzhalter für verschiedene Funktionen, von denen das System bei einem Aufruf der Operation abhängig vom Typ der Argumente eine auswählt und ausführt. Zum Beispiel kann die Operation `Size` mit verschiedensten mathematischen Objekten, etwa Halbgruppen, Gruppen, Konjugiertenklassen, Körpern aufgerufen werden, und sie soll die Anzahl der Elemente in diesen Strukturen zurückgeben.

Die Funktionen, die die eigentliche Arbeit tun, werden als *Methoden* einer Operation installiert. Bei der Installation einer Methode wird GAP mitgeteilt, welche Bedingungen die Argumente der Operation erfüllen müssen (technisch: welche Bits im Typ der Argumente gesetzt sein müssen), damit diese Funktion anwendbar ist. Wenn zum Beispiel `Size` mit einer Gruppe als Argument aufgerufen wird (ja, man kann auch `Order` sagen, das delegiert dann auf `Size`), dann hat GAP meist mehrere Methoden, die anwendbar sind. Etwa eine generische, die alle Elemente aufzählt, oder eine, die für Permutationsgruppen funktioniert und viel effizienter ist, oder eine, die die Antwort für symmetrische Gruppen liefert, ohne ein Gruppenelement zu benutzen, oder schließlich eine für Gruppen, in denen ihre Ordnung bereits gespeichert ist. Wenn GAP die Ordnung einmal berechnet hat, wird diese im Objekt gespeichert, und im Typ der Gruppe wird ein weiteres Bit gesetzt, das besagt, dass die Ordnung gespeichert ist.

Um in einem Fall, in dem mehrere Methoden anwendbar sind, zu entscheiden, welche Methode genommen werden soll, berechnet GAP zu jeder Methode einen *Rang*. Die grobe Idee ist hier, dass der Rang um so höher ist, je spezieller die Anforderung an die Argumente für die Anwendbarkeit sind. Hier wird davon ausgegangen, dass eine Methode für einen spezielleren Fall installiert wird, weil sie in diesem Fall effizienter ist.

Eine Stärke bei diesem Konzept ist, dass GAP-Objekte ihren Typ im Laufe einer GAP-Session ändern können und immer die zum Aufrufzeitpunkt besten Methoden ausgewählt werden können. Im oben genannten Beispiel muss höchstens ein Aufruf von `Size` mit einer Gruppe wirklich etwas rechnen, jeder weitere Aufruf ist dann sehr billig, weil immer die Methode genommen wird, die einfach den abgespeicherten Wert zurückgibt. Es gibt natürlich auch subtilere Beispiele, bei denen verschiedene Algorithmen konkurrieren.

Eine andere Stärke ist, dass diese Methodenauswahl auch für Operationen mit mehreren Argumenten funktioniert. Zum Beispiel gibt es viele Methoden für die Multiplikation zweier GAP-Objekte (ganze Zahl

mit ganzer Zahl, ganze Zahl mit Element aus endlichem Körper und umgekehrt, Permutation mit Permutation, Gruppenelement mit Untergruppe, ...), und weitere Fälle sind leicht durch die Installation weiterer Methoden hinzuzufügen, ohne dass irgendein existierender Code geändert werden muss. Ein anderes Beispiel ist `Centralizer`, aufrufbar mit einer Gruppe und einem anderen Argument. Hier gibt es verschiedene Methoden, wenn das zweite Argument ein Gruppenelement oder eine Untergruppe ist, oder es gibt eine spezielle Methode für den Fall, dass die Gruppe (weiß, dass sie) abelsch ist. Mehr Details sind in [BL] erklärt.

Organisation des Projektes

Die derzeitigen Entwickler von GAP sind über viele Länder verteilt. Die Kommunikation über Erweiterungen und Änderungen oder Korrekturen von GAP findet über Mailing-Listen statt. Das funktioniert ohne eine leitende Institution. Es sind lediglich vier GAP-Zentren benannt, die Infrastruktur für das Projekt bereitstellen und einen großen Teil der Wartungsarbeiten (Webseiten, Releases bereitstellen, Testen, Pakete aktualisieren, ...) übernehmen.

Für die Nutzer gibt es zwei Mail-Adressen, an die Fragen und Anmerkungen geschickt werden können, das *GAP-Forum* für Fragen von allgemeinem Interesse und eine Support-Liste für eher technische Fragen.

Es gibt eine *GAP-Support Group*, die versucht, Fragen an die genannten Listen möglichst zügig zu beantworten. Sie besteht aus den derzeitigen Entwicklern, Paket-Autoren sowie einigen engagierten Nutzern von GAP.

Schließlich möchte ich den *GAP-Council* erwähnen. Dessen Mitglieder stehen einerseits dem Projekt bei strategischen Fragen beratend zur Seite. Außerdem bilden sie aber auch das Editorial Board für eine weitere Besonderheit von GAP: Autoren von GAP-Paketen werden ermuntert, ihre Pakete zum *Referieren* einzureichen, analog zum Einreichen eines Artikels bei einer Fachzeitschrift. Unabhängige Gutachter sehen sich dann das Paket unter inhaltlichen, fachlichen und technischen Gesichtspunkten an. Das Referieren hat schon bei vielen Paketen zu nützlichen Verbesserungen geführt, und mit der Akzeptanz eines Paketes soll den Autoren eine Anerkennung der oft beträchtlichen Arbeit gegeben werden, die in der Entwicklung des Paketes steckt. Zur Zeit sind 46 der oben erwähnten 75 Pakete, die über die GAP-Webseite erhältlich sind, akzeptiert.

Literatur

- [SGP] H. U. Besche, B. Eick und E. A. O'Brien. The groups of order at most 2000. *Electron. Research Announc. Amer. Math. Soc.*, 7, 1–4, 2001.
- [BL] Th. Breuer und S. Linton. The GAP 4 Type System: Organising Algebraic Algorithms. *ISSAC Proceedings*, ACM, 1998.

[ATL] J. H. Conway, R. T. Curtis, R. A. Wilson, S. P. Norton und R. A. Parker. *ATLAS of Finite Groups*. Oxford University Press, 1985.

[GAP] The GAP Group, Aachen, St Andrews. *GAP – Groups, Algorithms, and Programming, Version 4.4.10*, 2007 (see <http://www.gap-system.org>).

[GPL] GNU General Public License (see <http://www.gnu.org/copyleft/gpl.html>).

[TGP] A. Hulpke. Constructing transitive permutation groups. *J. Symb. Comp.*, 39:1–30, 2005.

[MAT] C. Jansen, K. Lux, R. Parker und R. Wilson. An atlas of Brauer characters. *The Clarendon Press Oxford University Press, LMS Monographs. New Series*, 11, New York, 1995.

Neues aus Waterloo: Maple 12 und MapleSim 1.0

Scientific Computers GmbH (Aachen)

T.Richard@scientific.de



Das Jahr 2008 bringt neben der Version 12 von Maplesofts Kernprodukt Maple eine völlig neue Produktlinie: **MapleSim** – mehr dazu am Ende dieses Artikels.

Zunächst zu den wichtigsten Neuerungen in Maple selbst; aus Platzgründen sind nur einige Highlights herausgegriffen.

Grafik und GUI

Im Grafik-Bereich bietet der *Plotting Guide* als Teil der eingebauten Dokumentation einen direkten Weg von der visuellen Darstellung eines Sachverhalts (egal ob simpler Funktionsplot, Richtungsfeld, statistisches Diagramm o. Ä.) zur Hilfe-Seite des zugehörigen Kommandos, dessen Syntax und Optionsvielfalt sonst manchen Erstanwender abschrecken könnte. Der Befehl **plots[polarplot]** kennt nun echte polare Achsen, und ganz neu ist der Befehl **plots[dualaxisplot]**, mit dem erstmals zwei vertikale Achsen in 2D-Plots verwendet und unterschiedlich skaliert werden können – ein lange gehegter Wunsch vieler Benutzer. Daneben wurde die Syntax für Punkte-Plots deutlich vereinfacht. In der Plot-Komponente für interaktive Dokumente lassen sich jetzt Maus-Aktionen (*Klicken, Ziehen, Loslassen*) mit Reaktionen in Form von Befehlsaufrufen verbinden. Dies ermöglicht intuitivere grafische Applikationen, obwohl es sich vorerst auf den 2D-Fall beschränkt. Gleiches gilt für das Beibehalten von Plot-Attributen bei Wiederausführung eines Worksheets, wodurch viel Anpassungsarbeit entfällt. Mit dem CAD-Link lassen sich Parameter aus Computer-Aided-Design-Modellen extrahieren, verarbeiten und zurückschreiben. Unterstützt werden hierbei die 2008er-Versionen von Autodesk Inventor und SolidWorks unter Windows.

Symbolik

Das Unterpaket **ParametricPolynomialSystems** zum bekannten **RootFinding** behandelt Gleichungen und

Ungleichungen, die polynomial in zwei „Sorten“ von Variablen sind: in Unbestimmten und Parametern. Es liefert zum Beispiel die Anzahl von Lösungen in Abhängigkeit von Parametern oder beantwortet die Frage, unter welchen Bedingungen überhaupt Lösungen in einem vorgegebenen Gebiet existieren. Eine der Techniken, die dabei zum Einsatz kommt, ist Cylindrical Algebraic Decomposition – die Abkürzung CAD hat in Maple 12 also eine zweite Bedeutung.

Die Behandlung von Differentialgleichungen genießt traditionell einen hohen Stellenwert bei Maplesoft. Stellvertretend seien hier neue Algorithmen für bestimmte ODEs dritter Ordnung sowie die Methode der Laplaceschen Invarianten für PDEs zweiter Ordnung in zwei Unbekannten genannt.

In der linearen Algebra wurde ein Befehl für das Kronecker-Produkt von Matrizen hinzugefügt. Auch die großen Pakete **Physics** und **DifferentialGeometry**, die in Maple 11 hinzukamen, wurden ausgebaut. Details dazu finden sich wie immer in der Update-Dokumentation.

Numerik und Effizienz

Der Befehl **ifactor** zur Ganzzahl-Faktorisierung hat weitere Verfahren „gelernt“: MPQS (*MultiPolynomialQuadratic Sieve*) und die Kettenbruch-Methode von Morrison-Brillhart. Er verwendet per Default eine Kombination aus diesen beiden sowie dem Pollard-rho-Verfahren, wodurch viele Faktorisierungen spürbar schneller geworden sind.

Der Compiler für benutzerdefinierte numerische Prozeduren verarbeitet nun auch komplexe Zahlen (in arithmetischen Operationen und elementaren Funktionen), ebenso der Quelltextgenerator, sofern die Zielsprache diesen Datentyp kennt – das sind also Fortran und MATLAB. Für die letztgenannte Sprache kann Maple 12 sogar Quelltext importieren und in die eigene Syntax konvertieren. Dies geschieht mit Hilfe zusätzlicher Befehle im MATLAB-Paket. Der Mechanismus deckt zwar nicht den vollen MATLAB-Sprachumfang ab, ist

aber vom Anwender erweiterbar. Thematisch eng damit verbunden sind neue Sprach-Features in Maple, etwa das sogenannte *programmer indexing*, das einen flexibleren (weil näher an der internen Darstellung angelehnten) Zugang schafft und beispielsweise dynamisch wachsende Arrays erlaubt.

Das Paket **CurveFitting** hat einen Befehl **ArrayInterpolation** erhalten, der auf Effizienz bei mehrdimensionalen Datensätzen getrimmt ist und hauptsächlich für schnelles Resampling und Table Lookups dient. Per Default interpoliert er linear, je nach Option aber auch mittels verschiedener Splines und anderer Methoden (nearest, highest und lowest neighbour).

DynamicSystems ist ein Paket zur Analyse linearer zeitinvarianter Systeme. Damit kann der Regelungstechniker kontinuierliche und diskrete Systeme durch Differential- bzw. Differenzgleichungen sowie Übertragungsfunktionen, Zustandsraum-Matrizen, Pol-Nullstellen-Listen usw. beschreiben und auf vielfältige Weise untersuchen, etwa mit Bode-Plots, Wurzelortskurven, Amplitudengang, Steuerbarkeits- und Beobachtbarkeitsmatrizen. Diese Techniken waren bisher nur in der BlockBuilder-Toolbox verfügbar, die dem Export solcher Systembeschreibungen von Maple nach Simulink dient. Die **Database Toolbox** ist sogar komplett ins Hauptprodukt integriert worden, so dass man ohne Zusatzkosten auf relationale Datenbanken zugreifen kann (einen JDBC-Treiber vorausgesetzt).

Um diverse Kommandos zur Behandlung von Wavelets wurde das Paket **DiscreteTransforms** ergänzt: damit lassen sich die Koeffizienten für bestimmte Klassen von Wavelets (Daubechies, Coiflet, Symlet, Battle-Lemarie sowie die biorthogonalen Cohen-Daubechies-Feauveau und Splines) numerisch ermitteln, und **WaveletPlot** dient ihrer Visualisierung. In Verbindung mit **ImageTools** lassen sich so z. B. Wavelet-Transformationen und ihre Inverse auf Bilddaten durchführen, was in einem mitgelieferten Worksheet exemplarisch demonstriert wird.

Technisches

Neben Linux stehen endlich auch native 64-bit-Implementierungen für Windows und Mac OS X zur Verfügung; lediglich der Itanium-Prozessor wird nicht mehr unterstützt. Das Multithreading wurde leicht verbessert; so gibt es nun keine Unterscheidung mehr zwischen single-threaded und multithreaded Kernel, sondern der „wiedervereinigte“ Kernel bietet beides: per default arbeitet er single-threaded, und sobald man das Threads-Paket nutzt, schaltet er um.

MapleSim 1.0

MapleSim ist eine komplette Umgebung zur Modellierung und Simulation physikalischer Systeme, die aus so unterschiedlichen Bereichen („Domains“) wie Signalfluss, Elektrotechnik und Elektronik, 1D-Mechanik (rotatorisch und translatorisch), 3D-Mehrkörpermechanik und Thermodynamik zusammengesetzt sein können.

Insgesamt stehen ca. 450 Komponenten zur Verfügung, die sich aus hierarchisch geschachtelten Paletten auswählen lassen. Die Verbindung und Parametrisierung der Komponenten in einem Diagramm erfolgt sehr einfach und übersichtlich per Maus und Kontextmenü. Das Bildschirmfoto auf der nächsten Seite zeigt ein mitgeliefertes Beispiel, bei dem ein PID-Regler die Geschwindigkeit eines Gleichstrommotors samt nachgeschaltetem Getriebe (mit mechanischen Größen wie Trägheit und Spiel) kontrolliert.

Eine Besonderheit in MapleSim ist, dass sowohl kausale (d. h. signalfluss-orientierte) als auch akausale (topologische) Modellierung unterstützt wird, wobei letztere deutlich näher am physikalischen Vorbild liegt und mit weniger Komponenten auskommt als erstere. Die meisten Komponenten basieren auf Beschreibungen in der hersteller-übergreifenden Standardsprache *Modelica*, weshalb MapleSim auch entsprechenden Code exportieren kann – die entgegengesetzte Richtung wird in einer späteren Version zugänglich sein.

Eine Ausnahme ist die Mehrkörpermechanik, wofür MapleSim auf eine leistungsfähigere eigene Implementierung setzt, die dem bisherigen Produkt DynaFlexPro zugrundeliegt. Gegenüber Modelica bietet sie diverse Vorteile, u. A. ist man nicht auf absolute Koordinaten beschränkt.

Nach der Eingabe von Diagramm und Parametern sowie der Platzierung von *Probes* (abstrakten Messwert-Aufnehmern) klickt der Ingenieur auf den „Run Simulation“-Knopf, und MapleSim erstellt aus dem Netzwerk der Komponenten ein (üblicherweise differential-algebraisches) Gleichungssystem. Dieses wird mit Hilfe von Maple symbolisch vereinfacht und anschließend numerisch gelöst. Dabei ergeben sich drastische Geschwindigkeits-Steigerungen gegenüber rein numerisch arbeitender Software. Ein konkretes Problem aus der Simulation eines Verbrennungsmotors konnte so von 2300 auf 150 Gleichungen reduziert werden, was eine 10-fache Beschleunigung zur Folge hatte. Zudem halten die symbolischen Techniken die unvermeidlichen Rundungsfehler bei der Index-Reduktion von DAEs im akzeptablen Rahmen.

Die Resultate werden anschließend als 2D-Plots visualisiert und können auch als numerische Daten weiterverarbeitet und exportiert werden. Dabei ist jeder physikalischen Größe, die man in einer *Probe* angewählt hat, ein Plot zugeordnet.

Im Idealfall muss der Ingenieur also die MapleSim-Oberfläche gar nicht verlassen und bekommt so die Mathematik dahinter überhaupt nicht zu sehen. Zur vertieften Analyse sind jedoch Nachbearbeitungen der Modelle in sog. *Templates* möglich, das sind vorgefertigte Maple-Worksheets, die aus MapleSim heraus gestartet werden. Sie umfassen insbesondere die regelungstechnische Auslegung, die Extraktion von Gleichungen, Anfangswerten, Zuständen, Ein- und Ausgängen usw., die Parameter-Optimierung, die Daten- und C-Code-Generierung sowie das Erstellen benutzereigener Komponenten.

Ein wichtiges Feld ist die Dokumentation von Mo-

dellen: diese kann aus Dateien beliebiger Formate bestehen (Maple-Worksheets, Texte, Skizzen, Messwerte, etc.) und wird im MapleSim-eigenen Format *msim* wie in einem Container aufgenommen. Parametersätze lassen sich getrennt verwalten, und Kennlinien (Lookup-Tables) z. B. aus xls-Dateien entnehmen.

Von Maple hat die neue Umgebung außerdem die korrekte Behandlung physikalischer Einheiten geerbt, die automatisch in die Parameterliste übernommen und dadurch Teil der Dokumentation werden. Dies erlaubt eine Plausibilitätsprüfung als Bestandteil der Modellvalidierung.

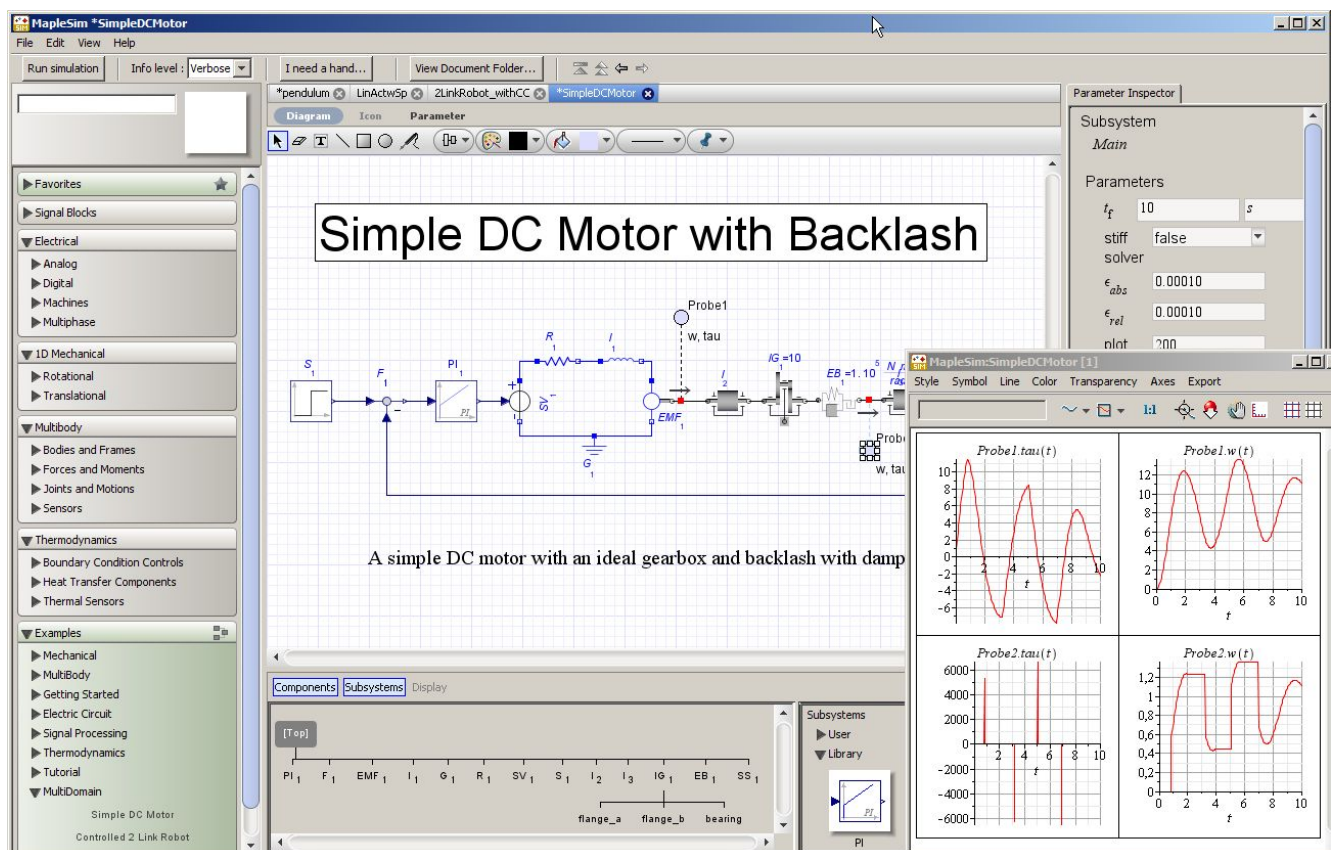
Der Benutzer kann ganze Bibliotheken eigener Modelle erstellen und diese mit anderen Anwendern austauschen.

Selbstverständlich kann man auf erstellte Modelle auch aus einem reinen Maple-Worksheet zugreifen, d. h. ohne die MapleSim-Oberfläche erneut zu starten. Mit dem gleichnamigen Paket lassen sich dann Simulationsläufe automatisieren, um beispielsweise umfangreiche Parameterstudien unbeaufsichtigt im Batchbetrieb durchzuführen.

Der Produkteinführung im Oktober ist ein mehrmonatiger Pilot-Test vorausgegangen, bei dem künftige Anwender wertvolles Feedback gegeben haben. Zu diesem neuartigen Produkt wird es in schneller Folge Toolboxes geben, die z. B. weitere physikalische Domains erschließen und die Verbindung zu Simulink herstellen. Auch die direkte Code-Generierung für Echtzeit-Plattformen (Hardware-in-the-Loop) steht auf der Agenda, ebenso wie die Einbindung von Modelica-Libraries von Drittherstellern. MapleSim ist also nicht einfach nur ein weiteres Add-On zu Maple, sondern der Ausgangspunkt einer ganzen Produktlinie, bei der die Computeralgebra neue Anwendungsgebiete erschließt. Aus technischer Sicht ist MapleSim aber durchaus eine Toolbox, setzt somit Maple 12.01 voraus und läuft auf den gleichen Plattformen.

Weitere Informationen

Mehr zum Thema findet sich wie üblich auf Deutsch unter <http://www.scientific.de> bzw. auf Englisch unter <http://www.maplesoft.com>.



Bildschirmfoto Gleichstrommotor

Der Schulversuch CALiMERO

Studienseminar Oldenburg

hen.koerner@t-online.de



Eine Kulturtechnik befördert die Leistungen der Intelligenz durch Versinnlichung und exteriorisierende Operationalisierung des Denkens. Das Kognitive bleibt nicht eingeschlossen in die unsichtbare Innerlichkeit der mentalen Zustände eines Individuums; Intelligenz und Geist werden zu einer Art distributivem, damit auch kollektivem Phänomen, das sich bildet im handgreiflichen Umgang des Menschen mit Dingen und symbolischen und technischen Artefakten.

S. Krämer/H. Bredekamp ([5], S. 18)

Dass der Umgang mit Symbolischem (Texten) zur Kulturtechnik gehört und die Leistungsfähigkeit des Menschen erhöht, ist wohl Allgemeingut; dass dies auch für den Umgang mit technischen Artefakten gilt, manchmal weniger. Gerade Mathematik wird dann als die Wissenschaft angesehen, in der die Technikfreiheit geradezu konstitutiv ist. Im niedersächsischen Schulversuch CALiMERO (ComputerAlgebra im Mathematikunterricht: Entdecken, Rechnen Organisieren) werden Möglichkeiten und Effekte des Einsatzes einer mächtigen Technik (CAS, hier: V200 von TI) im Unterricht untersucht. Dieses Projekt ist in mehrerer Hinsicht bedeutsam (vgl. auch Computeralgebra-Rundbrief 35, 2004, S. 29-30):

- (1) die große Anzahl an beteiligten Schülern und Lehrern (29 Schulklassen an 6 Gymnasien),
- (2) die gemeinsame Planung, Durchführung und Evaluation des Unterrichts durch die beteiligten Lehrer in Kooperation mit einer wissenschaftlichen Begleitung (Prof. Dr. Bruder),
- (3) der voll integrierte Einsatz des CAS in Lern- und Leistungssituationen in den Jahrgängen 7-10.

Zu (1): Der quantitative Umfang ermöglicht wohl zum ersten Mal umfangreiche und aussagekräftige empirisch fundierte Aussagen. Erste Ergebnisse liegen vor [1].

Zu (2): Längst nicht alle beteiligten Lehrer verfügten zu Beginn des Projekts über Unterrichtserfahrungen mit einem CAS. In den Versuchsschulen wird immer der gesamte Jahrgang mit dem CAS unterrichtet, so dass insgesamt eine gewisse Repräsentativität bezüglich der Unterrichtssituationen entsteht. Es sind nicht speziell motivierte Lehrer, die in ausgewählten Lerngruppen ausgesuchte Inhalte, womöglich noch in einem eng begrenzten Zeitrahmen, unterrichten, sondern große Teile von Fachgruppen, die die normalen Inhalte in ihrer ganzen Breite unterrichten, kurz: CAS wird zum normalen Werkzeug. Ein charakteristisches Element des Projekts sind viermal im Jahr stattfindende 3-tägige Arbeitstagungen der beteiligten Lehrer, in denen der Unterricht geplant und ausgewertet wird und Erfahrungen ausgetauscht werden. Didaktisch-methodische Impulse gibt es dabei von wissenschaftlicher Seite durch Prof. Dr. Bruder, eine fachbezogene durchgehende Begleitung und Unterstützung durch Fachleiter von Studienseminaren. Immer wieder geforderte Kooperation von Lehrkräften findet hier also in hohem Maße statt und ist konstitutiv für das Projekt. Im unmittelbar nachfolgenden Jahrgang werden die erstellten Materialien evaluiert, so dass sich ein reflexiver Zyklus aus Planung-Durchführung-Evaluation-Modifikation-Durchführung-Evaluation ergibt. Die sich dabei im Wechselspiel aus Planung, Durchführung und Evaluation ergebenden Modifikationen in den Einstellungen und Vorstellungen der beteiligten Lehrer gehören zu den spannenden Aspekten des Projekts. Sie werden, zumindest teilweise, im Heft 4/2009 von *Der Mathematikunterricht* dargestellt werden.

Zu (3): Im Projekt wird ein durchgängiges didaktisch-methodisches Gesamtkonzept eines Mathematikunterrichts mit einem CAS erarbeitet (Teile veröffentlicht in [2], [3]). Es handelt sich damit insofern um ein ganzheitliches Konzept als alle Standardinhalte und prozessorientierten Kompetenzen zu berücksichtigen sind und damit Inhalte und Verfahren, die ohne CAS zu bearbeiten sind, ebenso im Blick stehen wie die konkreten Einsatzmöglichkeiten des CAS. Es werden also auch Übungsphasen gestaltet, Sicherungen des Wissens in den Blick genom-

men und adäquate Leistungsüberprüfungen geschaffen. Natürlich und unumgänglich ist damit auch ständig die Frage nach Umfang und Tiefe händischer Fähigkeiten und Fertigkeiten virulent. Da zu dieser Frage keine sicheren empirisch geprüften Aussagen vorliegen, sind hier Setzungen vorgenommen worden. So werden zunächst alle neuen Algorithmen ohne CAS eingeführt und mit einfachen Eingaben und Koeffizienten geübt. Dies gilt auch für Termumformungen, wo alle kanonischen Umformungen (Zusammenfassen, Ausmultiplizieren, Ausklammern) händisch geübt werden, aber nicht in Verschachtelungen (mehrere Klammern, Doppelbrüche etc.). Die geforderten händischen Fähigkeiten werden durchgehend explizit in den einzelnen Themenbereichen angegeben. Insgesamt wird darüberhinaus dem Kopfrechnen ein hoher Stellenwert eingeräumt. So finden regelmäßige (wöchentlich) Kopfübungen statt, in denen grundlegende Fertigkeiten zu unterrichtlich zurückliegenden Inhalten systematisch gefestigt werden. Thesenhaft formuliert: Je mehr an Technik delegiert wird, desto mehr muss – in einfachen Ausformungen – händisch sicher beherrscht werden. Dahinter steckt die Annahme, dass sicherer Umgang mit einem CAS grundlegende händische Fertigkeiten voraussetzt, dass dies vor allem auch für die wichtige „Strukturerkennungskompetenz“ gilt. Im Vergleich zum bisherigen Unterricht ergibt sich damit wohl ein Mehr an Kopfrechnen mit einfachen Werten und ein Weniger an händischen Übungen auf mittlerem Komplexitätsniveau, wie sie im CAS-freien Unterricht vorherrschen. Neben diesen inhaltlichen Verschiebungen im Verhältnis von „händisch“ und „mit Technik“ tritt eine weitere Modifikation in der inhaltsbezogenen Strukturierung, die teilweise auch durch den CAS-Einsatz bedingt ist, im Übrigen aber Folge der zeitlichen Verdichtung durch G8 ist. Viele Themenbereiche werden zunächst in grundlegender Weise händisch oder mit CAS eingeführt, aber häufig nicht bis zur Routinisierung geübt, sondern an möglichst vielen Stellen wieder aufgenommen und weiter bearbeitet, in der Hoffnung, dass die höhere Frequenz unterrichtlichen Auftretens von Inhalten die fehlende Übungstiefe mehr als kompensiert. Die Darstellungen unten ((A) und (B)) konkretisieren dies.

Die weiteren Ausführungen hier beschränken sich auf (3) und dort weitgehend auf die inhaltlichen Auswirkungen, die ein voll integrierter CAS-Einsatz hat.

Unterricht mit CAS: Was bleibt? Was ändert sich? Die Sache ist kompliziert: Es geht einerseits primär um Möglichkeiten, mit Lerngruppen produktiv und ertragreich Mathematik zu betreiben. Die Frage nach dem CAS-Einsatz ist dann immer eine nachgeordnete, sie hat dienende Funktion im Sinne der primären didaktischen Ziele, CAS ist dann „nur“ ein Hilfsmittel, ein Werkzeug. Andererseits gehört Technik und ihr Gebrauch konstitutiv zum Menschen und verändert seinen Zugang zur Welt und Umgang mit ihr. Aus diesem Blickwinkel heraus ist zu fragen, wie weit sich Inhalte, Schwerpunkte und Methoden des Mathematikunterrichts in Folge eines so mächtigen Werkzeuges wie ein CAS es ist ändern

können, sollen oder müssen. Im Folgenden werden die Aspekte in den Fokus genommen, die spezifisch für ein CAS sind. So werden im Projekt natürlich auch die grafisch-numerischen Möglichkeiten immer wieder so weit genutzt, dass die Trias Tabelle – Grafik – Term die traditionelle Überbetonung der Algebra und vor allem die damit einhergehende Dominanz der syntaktischen Anteile ersetzt. Da diese folgenreiche Modifikation aber auch schon durch GTR und TK erfolgt, wird sie hier nicht weiter ausgeführt, muss aber als ein zentraler Baustein ständig präsent sein.

Es sind zwei Aspekte, die die grundlegenden Erweiterungen des CAS gegenüber allein grafisch-numerischen Werkzeugen ausmachen:

- (A) Freie Definition von – mehrstelligen – Funktionen.
- (B) Syntaktische (algebraische) Fertigkeiten des CAS.

(A) Ein grundlegendes Merkmal von CAS ist die Möglichkeit, Formeln als Funktionen zu definieren und diese dann als Makros in Problembearbeitungen einzusetzen. Weiterhin schafft die freie Benennbarkeit der Variablen und Funktionen darüber hinaus die Möglichkeit, den Übergang von Alltagssprachlichen Formulierungen über „Wortformeln“ zu formalen Darstellungen feiner zu diskretisieren.

Beispiel:

- (1) Faustregel: Der Bremsweg hängt von der Geschwindigkeit ab; er ist Geschwindigkeit mal Geschwindigkeit dividiert durch 100.
- (2) $\text{Bremsweg} = \frac{\text{Geschwindigkeit} \cdot \text{Geschwindigkeit}}{100}$
- (3) $\text{brems}(\text{gesch}) = \frac{\text{gesch} \cdot \text{gesch}}{100}$
- (4) $\text{brems}(v) = \frac{v \cdot v}{100}$

(2), (3) und (4) sind Möglichkeiten der Eingabe der Faustregel als Formel in das CAS. Wenn Formeln auf diese Weise schon früh auch als Funktionen eingeführt werden, findet die wichtige Vernetzung von Formeln und Funktionen (Zuordnungen) entsprechend frühzeitig und explizit statt. Meist kennen Schüler schon Formeln (z. B. Umfang/Fläche eines Rechtecks) und auch Zuordnungen. Da dies aber in unterschiedlichen Kontexten (Geometrie-Zuordnungen) stattfindet, wird die inhaltliche Beziehung häufig wenig gesehen. Dies zeigt sich später dann deutlich, wenn man Schüler fragt, was die Formel für den Flächeninhalt eines Kreises mit Parabeln zu tun hat. Wenn man $F(r) = \pi r^2$ statt $F = \pi r^2$ lernt, ist der Zusammenhang augenscheinlich und heuristisch produktiv, wenn es z. B. um Kovarianz von Radius und Flächeninhalt geht. Da diese Zusammenhänge aber eben nicht selbsterklärend sind, müssen sie aktiv herbeigeführt werden und behutsam aber systematisch eingeführt werden. Im Projekt beginnt die Einführung in Klasse 7 im Zusammenhang mit dem Kennenlernen und Auswerten von Formeln für Zuordnungen. Mit dem

BMI (BodyMassIndex) kann hier dann auch schon eine mehrstellige Funktion eingeführt werden (vgl. [4]). Werden die Definitionen von Funktionen im Themenbereich „Zuordnungen“ eingeführt, so werden sie im Themenbereich „Flächenbestimmung“ wieder aufgenommen [2], Bd.1, S.49f.:

Für die Trapezfläche wird folgende Gleichung allgemein formuliert:

$$atrapez(a, c, h) = \frac{1}{2}(a + c) \cdot h.$$

- Gib die Formeln zu den unten stehenden Eintragungen (1), (2) und (3) an. Skizziere zu jeder Formel drei Trapeze. Beschreibe, wie die Trapeze sich ändern, wenn man x ändert.
 (1) $atrapez(6, 4, x)$
 (2) $atrapez(6, x, 2)$
 (3) $atrapez(x, 4, 2)$
- Erkläre den Fall $x = 0$ auch geometrisch.
- Skizziere jeweils für (1), (2) und (3) die Zuordnung $x \mapsto \text{Trapezfläche}$ in ein Koordinatensystem. Erkläre die Bedeutung der Schnittpunkte.

Das Anwenden von Formeln für Flächeninhalte wird durch funktionale Betrachtungen und entsprechende Implementierung im CAS ergänzt, das CAS wird zur dynamischen Formelsammlung. Neben die klassische Darstellung der Formel in ihrer syntaktischen Struktur tritt die Darstellung als Funktion ($\frac{1}{2}gh \leftrightarrow \text{adriereck}(g, h)$). Letztere ermöglicht dann Kovarianzuntersuchungen mit entsprechenden grafisch-tabellarischen Verfahren indem ein Argument mit „ x “ bezeichnet wird. Gelegentlich wird der funktionalen Schreibweise die mathematische Dignität abgesprochen (in Prüfungen ist sie dann sogar verboten), dies ist aber grundsätzlich falsch, weil beide Darstellungen ihre je spezifischen Möglichkeiten und natürlich auch Grenzen haben. Das Substituieren einzelner Argumente durch Terme und die Untersuchung von Auswirkungen von Parametervariationen lassen sich mit der Funktionsschreibweise übersichtlich gestalten und durchführen, die innere Struktur einer Formel bedarf dagegen natürlich der entsprechenden syntaktischen Darstellung. Wichtig für einen verständnisvollen Umgang mit solchen Formeln sind Übungen zur inhaltlichen Rückinterpretation, also z. B.: *Welche Frage kann mit $BMI(70, x)$ beantwortet werden?* (weitere Beispiele in [2], Bd.1, S.49, Aufgabe 1a, 3), und zum aktiven Wechsel der Darstellungsformen (a. a. O. Aufgabe 4a). Im Themenbereich „Satz des Pythagoras“ können Schüler dann selbstständig Funktionen zur Berechnung fehlender Größen erzeugen ($lang(a, b) = \sqrt{a^2 + b^2}$, $kurz(a, c) = \sqrt{c^2 - a^2}$) und dann auch wieder in wechselnden Kontexten funktionale Zusammenhänge erforschen, z. B. $kurz(5, x)$ bzw. $kurz(x, 25)$.

Auch bei Erschließung und Untersuchung von Funktionsklassen kann die funktionale Darstellung benutzt werden, auch wenn diese hier keinen zwingenden Vorteil hat. Es werden aber immer wieder der Umgang mit dieser Darstellungsform und die wechselseitigen Bezüge aus Funktionsaufruf, inhaltlicher Bedeutung und syntaktischer Struktur geübt und gefestigt.

Wir bauen ein Potenzfunktionenmakro:

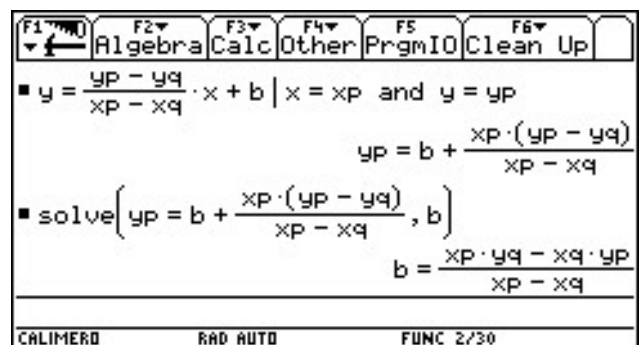
$$pot(x, k, a, b, c) = a \cdot (x - b)^k + c$$

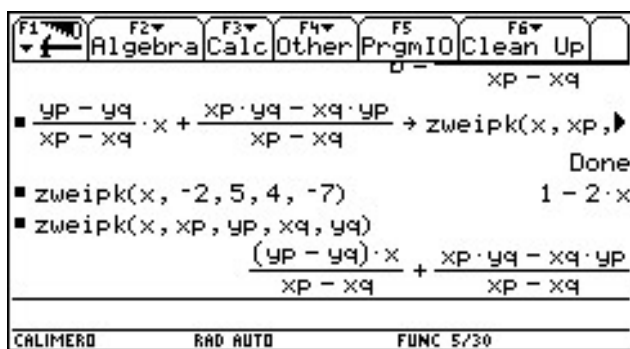
- Beschreibe, welche Bedeutung die einzelnen Parameter haben.
- Mit welcher Eingabe wird $f(x) = 2 \cdot (x - 5)^4 - 8$ gebaut?
 - Was bedeutet $pot(1, 3, 1, -2, 5)$? Stelle eine Frage, für deren Beantwortung diese Eingabe eine Lösung liefert.
 - Was musst du eingeben um die Grundfunktionen $f(x, n) = x^n$ zu erhalten?
 - Mit welcher Eingabe kannst du den Funktionswert an der Stelle 3 von $f(x) = 24x^{-3} + \frac{1}{4}$ berechnen? Überprüfe dein Ergebnis durch eine Rechnung „zu Fuß“.
 - Was bedeutet $pot(3, 4, 1, 0, c)$? Stelle eine Frage, für deren Beantwortung diese Eingabe eine Lösung liefert.
 - Wie kann man mit pot die Schnittstellen mit der y -Achse bestimmen?

Insgesamt soll so im Verlauf der Klassen 7 bis 10 an vielen Stellen, aufeinander aufbauend, thematisiert in verschiedenen Kontexten, eine Kompetenz im funktionalen Umgang mit Formeln erzeugt werden, so dass Schüler am Ende selbstständig, je nach Bedarf, entsprechende Funktionen individuell erzeugen und zur Problembearbeitung einsetzen können.

Darüber hinaus entsteht im CAS eine Formelsammlung, die den dynamisch, situativen Gebrauch der Formeln gestattet und sie damit zu einem produktiven Werkzeug machen. Die eigene „Programmierung“ solcher Formeln, z. B. die Flächen- und Volumenformeln, später vielleicht auch Abstandsformeln, setzt aber immer die algebraische Herleitung mit Variablen voraus, so dass Schüler hier auch unmittelbar die Produktivität und den Mehrwert des Arbeitens mit Formeln erleben können.

Umgekehrt gilt dasselbe: Wenn das Bestimmen der Gleichung einer Geraden durch zwei Punkte an konkreten Beispielen eingeführt und geübt wird, kann durch eine allgemeine Behandlung nach gleichem Schema ein entsprechendes Makro erstellt werden. Auch wenn dies wohl nur selten von Schülern in Klasse 7/8 selbstständig hergeleitet werden kann, so kann ein Nachvollzug mit anschließendem Umgang mit dem Makro sehr wohl eine produktive Einsicht und Vorbereitung für spätere selbstständige Entwicklungen leisten.





Solchermaßen programmierte Geräte werden damit lerngruppenspezifisch konfiguriert und ermöglichen darüber hinaus auch eine individualisierte Nutzung. Grundlegende Fähigkeit beim Arbeiten mit solchen selbst definierten Makros ist die Substitution, die im Grunde schon beim Arbeiten mit den eingebauten Befehlen erforderlich ist. Diese sind eben auch mehrstellige Funktionen, bei deren Anwendungen vorgegebene Argumente zielgerecht substituiert werden müssen (*solve(Gleichung, variable)*).

Unabhängig von allen inhaltsbezogenen Argumenten für das Arbeiten mit selbstdefinierten, mehrstelligen Funktionen liegt ein weiteres Argument für die Thematisierung dieser Aspekte in der Aufklärung über die Arbeitsweise benutzter Technik. Im CAS gibt es vielfältige, entsprechend vordefinierte Funktionen, deren Erzeugung auf diese Weise durchsichtig gemacht wird. *DOTP(Vektor1, Vektor2)* (Skalarprodukt) kann ein Schüler auf einfache Weise dann selbst erzeugen.

(B) Die algebraischen Fertigkeiten eines CAS führen zu einem qualitativen und quantitativen Wandel in der Bedeutung und Funktion händischer Verfahren. Zunächst können mangelnde händische Fertigkeiten damit natürlich ersetzt werden oder zur nachträglichen Überprüfung händischer Lösungen benutzt werden. Wenn ein CAS aber allein als Reparaturbetrieb mangelnder händischer Fähigkeiten und Fertigkeiten angesehen werden würde, würde sein Potenzial bezüglich inhaltlicher und prozessorientierter Kompetenzen weit unterschätzt werden.

Umgekehrt könnte man vordergründig im Erstzugriff auf händische Verfahren vielleicht sogar verzichten. Damit würde aber meist jegliche Einsicht in zugrunde liegende Verfahren verhindert. Genau hier liegt der Bedeutungswandel händischer Verfahren. Sie dienen weniger dazu, eine Lösung zu finden, sondern sollen vielmehr Einsicht in prinzipielle Lösungsmöglichkeiten geben und Transparenz schaffen. Das CAS bietet die Möglichkeit, Kompetenzen im Wechselspiel aus „Technikeinsatz“ und „Lösung von Hand“ zu erzeugen, eine undialektische Gegenüberstellung von „händisch“ und „mit Technik“ greift zu kurz und verspielt produktive Möglichkeiten. Dies soll am Beispiel des Lösens linearer Gleichungssysteme veranschaulicht werden.

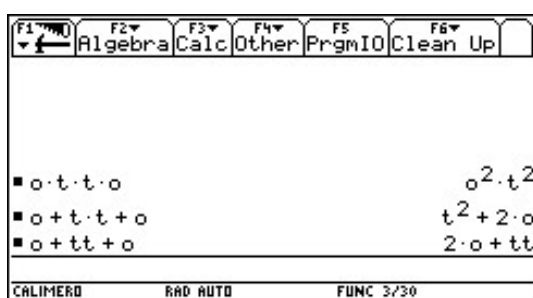
Ein CAS kann ein lineares Gleichungssystem durch direkte Eingabe der Gleichungen lösen. Dies wird im Projekt auch entsprechend zu Beginn eingeführt. Hier

wird die Lösungskompetenz dann zunächst vollständig an den TC übergeben. Der Preis ist fehlende Einsicht in zugrunde liegende Verfahren und Prinzipien. Es liegt also reines, technologiegestütztes, „know how“ vor. Um Einsicht zu bekommen, werden die Gleichungen dann als Geradengleichungen interpretiert und nach y aufgelöst.

Die grundsätzliche Einsicht, das auf Descartes und Fermat zurückgehende Prinzip der Analytischen Geometrie, ist sicher nicht an Technik delegierbar, sondern eine menschliche Denkleistung. Die Umsetzung kann aber wieder mit technologischer Unterstützung erfolgen (Auflösen nach y , Gleichsetzen der Terme, grafische Darstellung der Geraden), „know what“ rückt in den Mittelpunkt. Im Projekt wird, aus curricularen Gründen, das Gleichsetzungsverfahren mit dann auch möglichen grafischen Lösungen als alleiniges algebraisches Verfahren eingeführt und an einfachen Beispielen geübt. Es ist eben nicht mehr Ziel, Schülern möglichst vielfältige händische Verfahren zur Verfügung zu stellen („Gleich-, Einsetzungs-, Additionsverfahren“), damit Gleichungssysteme geschickt und ökonomisch gelöst werden können, sondern Ziel ist Einsicht in prinzipielle Lösbarkeit und dazu reicht ein Verfahren aus, die Knopfdrucklösung des CAS ist entmystifiziert. Übungen, die Einsicht fördern, gewinnen dann ebenso an Stellenwert wie Modellierungen (vgl. [2], Bd. 3, S.27/28).

Beim später erfolgenden Übergang zu drei Gleichungen kann zunächst mit dem gleichen Verfahren, auch wieder rechnerunterstützt, gearbeitet werden. Das Erleben recht umständlichen Vorgehens beim Auflösen nach y und anschließendem Gleichsetzens führt dann zur Einführung des Additionsverfahrens, das zunächst auch von Hand ein- und durchgeführt wird. Systematisierung führt zum Gauss-Verfahren und der Einführung der Matrix-Schreibweise; der vielleicht schon vorher benutzte Befehl *RREF* (Diagonalisierung, Zeilenreduzierung) wird strukturell einsichtig, aus der „black box“ wird eine „white box“ bzw. „grey box“. Später können dann algebraische (Rang einer Matrix, etc.) und geometrische (Ebenen Schnitte) Betrachtungen weitere Einsicht in Strukturen und Zusammenhänge liefern und mit Hilfe von Matrizen dann auch größere LGS wiederum mit Technik gelöst werden. Die Herausbildung der Kompetenz „Lösen eines Linearen Gleichungssystems (LGS)“ geschieht in wechselseitigem „Dialog“ mit dem CAS; es entsteht eine Art Spiralcurriculum, bei dem das CAS immer wieder auch Ausgangspunkt für die Weiterentwicklung von Verfahren sein kann. Das Interpretieren, Bewerten und Einsicht Gewinnen rückt in das Zentrum des mathematischen Handelns, das konkrete Ausrechnen rückt in den Hintergrund.

Die algebraischen Fertigkeiten eines CAS lassen sich auch für einen kreativen, schülerorientierten Zugang bzw. eine Festigung der algebraischen Rechengesetze nutzen (vgl. [2], Bd.1, S.48).



- Variiere die Eingabe des Namens *Otto* mit verschiedenen Rechenzeichen. Finde einen Eingabeterm, bei dem sich besonders viel verändert.
- Erkläre für zwei deiner Variationen, welche Rechengesetze angewendet wurden.
- Warum ist das „tt“ nicht noch weiter vereinfacht worden? Erkläre!

Wie oben ausgeführt, sind die hier dargelegten Möglichkeiten des CAS-Einsatzes im Schulversuch *CALiMERO* immer eingebettet in ein konzeptionelles Ganzes. Ziel ist die allgemeine Verbesserung der mathematischen Problemlösefähigkeit, und dazu gehören dann auch reflexive, zusammenfassende, sinnstiftende Elemente, wie sie Mindmaps, rückblickende Analysen, Selbstdiagnosen usw. darstellen, die konsequent und durchgängig in den Unterrichtsgang integriert sind. Darüber hinaus aber gilt: Wenn CAS-typische Fähigkeiten (frei definierbare Funktionen, syntaktische Umformungen) frühzeitig, konsequent und altersangemessen in die Erarbeitung und Festigung mathematischer Inhalte im Unterricht integriert werden, dann wird das CAS zu einem Werkzeug, bei dem die Art der Verwendung sowie die Kenntnisse und Fähigkeiten des Nutzers bezüglich des Geräts Verwendungszusammenhänge definieren und mögliche Breite und Tiefe der Nutzung festlegen. Bei der Integration in den Unterricht geht es dann zentral um die Entwicklung einer instrumentellen Wechselbeziehung zwischen Schüler und Computerwerkzeug. Kompetenzen entwickeln sich dabei in der Auseinandersetzung mit dem Werkzeug, indem immer wieder bestimmte Teilaspekte an die-

ses ausgelagert werden. So werden in den Materialien auch konsequenterweise die notwendigen technischen Fähigkeiten durchgehend in allen Themenbereichen explizit angegeben. Zur Kompetenz eines Schülers gehört dann zwangsweise auch die Kompetenz, Teile an das Computerwerkzeug zu übergeben. Man kann hier dann von „technologischer Kompetenz“ in dem Sinne sprechen, dass inhalts- bzw. prozessorientierte Kompetenzen im Dialog mit Computerwerkzeugen erworben werden, pointiert und etwas provokant ausgedrückt: Das CAS wird zu einem ausgelagerten, materialisierten Teil des Gehirns, mathematische Kompetenz beinhaltet auch die Benutzung von Technik:

Denken ... ist dann nicht mehr im Subjekt lokalisiert, sondern das System aus Subjekt und Kontext (das sind insbesondere die dort verfügbaren materiellen und mentalen Werkzeuge und Technologien) realisiert „Denkprozesse“.

W. Dörfler ([6], S. 37)

Literatur

- [1] R. Bruder. Bergfest für *CALiMERO* in Bergkirchen, TI-Nachrichten 1/2008, Sonderausgabe, S. 10-13.
- [2] R. Bruder/W. Weiskirch. *CALiMERO*, Arbeitsmaterialien Bd. 1-3, Münster 2007/2008.
- [3] R. Bruder/W. Weiskirch. *CALiMERO*, Methodische und didaktische Handreichung Bd. 1-2, Münster 2007/2008.
- [4] H. Körner. Bremsweg und BMI, in : TI-Nachrichten 1/2006, S. 17-22.
- [5] S. Krämer/H. Bredekamp (Hrsg.), Bild-Schrift-Zahl, München 2003.
- [6] H. G. Weigand/T. Weth. Computer im Mathematikunterricht, Heidelberg, Berlin 2002.

mathemas ordinate  www.ordinate.de

☎ 0431 23745-00/ ☒ -01 , info@ordinate.de → Software for mathematical people !

 **Mathematica, ExtendSim,**

MathType, KaleidaGraph, Fortran, NSBasic, @Risk

und a.m.

$\infty + \mu < \heartsuit$

$$\int_{x_1}^{x_2} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2} dx$$

mathemas ordinate, Dipl. Math. Carsten Herrmann, M. Sc.
Königsbergerstr. 97, 24161 Altenholz

Mehr als 20 Jahre Erfahrung mit Software-Distribution !

Integration der Nutzung eines CAS in der Veranstaltung „Elementare Analysis“

Frauke Arndt
Technische Universität Dortmund

Hans-Wolfgang Henn
Technische Universität Dortmund

`frauke.arndt@math.tu-dortmund.de`
`henn@math.tu-dortmund.de`



Lehrerausbildung in Dortmund

Im Jahr 2003 wurde in NRW eine neue Landeslehrerprüfungsordnung (LPO 2003) erlassen, in deren Folge wir in Dortmund alle Lehramtsstudiengänge (von Grundschule bis Gymnasien und Berufskolleg) in einem innovativen Modell umgestaltet haben [1]. Wenige Jahre danach wurde in Dortmund einer der vier NRW-Modellversuche zur Lehrerausbildung im BA/MA-Modell eingeführt.

Die in dem komplizierten Dortmunder Modell durch Einflussfaktoren, die außerhalb der Mathematik liegen, eingehenden „Visionen“ stehen unserem Eindruck nach manchmal im Widerspruch zur fachlichen Einsicht. Wir müssen also Studierenden nach der alten Prüfungsordnung von 1994, der neuen von 2003 und der Dortmunder BA/MA-Prüfungsordnung gerecht werden.

Die derzeit laufende Einführung einer gemeinsamen BA/MA-Studienordnung für alle Lehramtsstudiengänge in NRW, die auch für uns in Dortmund die Erstellung neuer Studienordnungen bedeutet, macht unsere Probleme nicht kleiner.

Computereinsatz in der Ausbildung

Es war von Anfang ein Ziel bei unseren neuen Studienordnungen, in den Lehrveranstaltungen für alle Lehramtsstudierende mathematische Software, genauer gesagt Tabellenkalkulation, dynamische Geometriesoftware und Computeralgebrasysteme einzubeziehen.

In diesem Beitrag berichten wir über die „Elementare Analysis“, eine aus zwei Vorlesungsstunden (verantwortlich Henn) und zwei Übungsstunden (verantwortlich Arndt) bestehende Veranstaltung für künftige Haupt-, Real- und Sonderschullehrer.

Diese Analysis ist eine Pflichtveranstaltung nach LPO 2003, eine Wahlpflichtveranstaltung in der BA-Studienordnung. Erfahrungsgemäß stellt diese Veranstaltung für die Studierenden eine besondere Hürde

dar. Die Vorlesung geht zwar inhaltlich kaum über schulisches Grundkursniveau hinaus, jedoch liegt im Gegensatz zur Schule der Schwerpunkt auf einem verständnisorientierten Aufbau der Analysis einer Variablen, ausgehend vom Grenzwertbegriff, und nicht auf kalkülorientiertem Rechnen.

Der Einsatz eines CAS sollte in Schule und Universität selbstverständlich sein: Zum einen ist ein CAS ein leistungsfähiges Werkzeug zur Unterstützung von Modellbildungen und Simulationen, zum anderen kann ein CAS – vor allem durch dynamische Visualisierungen – den Aufbau adäquater Grundvorstellungen mathematischer Begriffe und Ergebnisse positiv beeinflussen, und schließlich befähigt der Computer durch die Möglichkeit heuristisch-experimentellen Arbeitens beim Problemlösen die dritte Grunderfahrung. Ein Lehrer, der seine Schülerinnen und Schüler zum sinnvollen Einsatz eines CAS anleiten will, muss jedoch an der Universität in gleichem Sinne mit einem CAS gearbeitet haben.

In der „Elementaren Analysis“ im Sommersemester 2007 haben wir konsequent versucht, zum Begriffsaufbau, zum Problemlösen und zum konstruktiven Üben ein CAS einzusetzen. Da uns das üblicherweise in unserer Fakultät eingesetzte CAS Maple für unsere Zielgruppe zu komplex (und viel zu teuer) erschien und da das bisher verwendete CAS Derive nicht mehr angeboten und gepflegt wird, haben wir uns für das kostenfreie CAS Maxima¹ (in der Version 5.15.0) entschieden und möchten mit diesem Beitrag ein Plädoyer für einen integrierten Umgang mit Maxima bei der Fachausbildung Analysis der Lehramtsstudierenden im SI-Bereich abgeben. Unter „integriert“ verstehen wir, dass keine zusätzliche Veranstaltung zum Erlernen des Programms und seiner Möglichkeiten angeboten wurde, sondern dass in der mathematischen Fachvorlesung „Elementare Analysis“ das CAS als selbstverständliche und sinnvolle Ergänzung auf einem einfachen Niveau regelmäßig genutzt wurde.

¹Erhältlich unter <http://maxima.sourceforge.net/>

Befragung von Studierenden des gymnasialen Lehramts

Um einen Überblick über den Computereinsatz in der Schule zu bekommen, wurden die 88 Studierenden des gymnasialen Lehramts², die im Frühjahr 2008 in der Vorlesung „Analysis II für Lehramt“ der TU Dortmund saßen, befragt. Auf die Frage, ob ein Computerprogramm in der Analysis in der Oberstufe genutzt wurde, antworteten nur 11 der Befragten mit „ja“, 14 weitere mit „eher ja“. 57 Studierende antworteten mit „nein“ oder „eher nein“. Gar keine Antwort gaben 6 Personen. 13 Studierende antworteten, dass sie in der Oberstufe weder Computer noch Taschenrechner benutzt haben.

In der Vorlesung „Analysis II für Lehramt“ wurden (im Gegensatz zu dem, was unsere entsprechende Studienordnung eigentlich vorschreibt) keine CAS eingesetzt. Aber immerhin 13 der Studierenden haben von sich aus einen Computer zur Übung oder Visualisierung genutzt. 72 Studierende haben (eher) kein Computerprogramm genutzt. 3 enthielten sich.

Frappierend an den oben genannten Zahlen ist, dass in der Oberstufe 25 der Befragten einen Computer (für die Analysis) nutzten, aber in der anschließenden Vorlesung nur noch 13. Aber Visualisierung hört in der Analysisveranstaltung der Universität nicht auf. Vielleicht setzen manche Dozenten implizit voraus, dass die Studierenden in der Lage sind, sich diese Anschauung selbst zu „beschaffen“, ob nun mit Computereinsatz oder ohne; eine Annahme, die sicher zu optimistisch ist.

Für den Analysisunterricht, den sie in Zukunft als Lehrerinnen und Lehrer geben wollen, geben dann auch nur 5 der Studierenden an, dass sie „mehr Visualisierungen“ und „mehr Computereinsatz“ machen wollen als ihre eigenen Lehrer. Einer der Studierenden schlägt vor, dass man das in der Didaktikveranstaltung thematisieren könne.

Konstruktives Arbeiten mit einem CAS in der Elementaren Analysis

Wie integriert man aber ein CAS in eine Veranstaltung? In unserem Fall war die Visualisierung mittels CAS in der Vorlesung „Elementare Analysis“ schon immer ein wichtiger Teil gewesen. Durch die Trennung von „Vorlesung“ und „Übungen“ war es aber in der Regel bei einer passiven Rezeption des in der Vorlesung Gebotenen durch die Studierenden geblieben.

- Wie bekommt man die Studierenden „an den Rechner“, obwohl in der Klausur kein Rechner genutzt werden kann und obwohl im Übungsraum keine Computer zur Verfügung stehen?
- Und wie vermittelt man den Studierenden, dass die Nutzung des CAS nicht *das* Ziel der Veranstaltung ist, sondern dass es ein unabdingbares, selbstverständliches Werkzeug bei der eigenen Arbeit wird?

- Wie geht man mit Studierenden um, die in der Schulzeit weder mit Computer noch mit CA-Taschenrechner regelmäßig gearbeitet haben?

Wir haben uns bemüht, den Übergang für die Studierenden so glatt wie möglich zu gestalten. Um sie „an den Computer“ zu bekommen, wurden zunächst alle notwendigen Voraussetzungen geschaffen: Da an der Raumsituation nichts zu ändern war, musste allen Studierenden ein entsprechendes CAS für daheim zur Verfügung gestellt werden. Wir entschieden uns für Maxima, da dieses CAS einerseits kostenlos und andererseits recht einfach zu bedienen ist. Eine zusätzlich eingerichtete „Computersprechstunde“ stellte sicher, dass die Studierenden technische und inhaltliche Fragen stellen konnten und auch, dass die wenigen, die keine Möglichkeit hatten, Maxima zu Hause zu nutzen, zu bestimmten Zeiten in der Universität ihre Aufgaben erledigen konnten.

Schließlich galt es, entsprechende Aufgaben in das bestehende Übungskonzept zu integrieren, und zwar so, dass Einführung und Nutzung des CAS an keiner Stelle zu eigenen Lernzielen wurden, sondern stets nur wichtige Lernziele der Analysis unterstützten. Maxima soll als sinnvolles, selbstverständliches und ergänzendes Werkzeug empfunden werden, das die Anschauung stärken und einem das Rechnen erleichtern kann.

Um die Nutzung auf selbständiger Basis zu etablieren, haben wir uns dazu entschieden, in jeder Woche eine „Maxima-Aufgabe“ zu stellen. In den Übungen konnten die Studierenden ihre Lösungen mit Hilfe eines Laptops mit Beamer präsentieren.

Beispiele

Etwa 8 der 13 Übungsblätter enthielten kleinere Anleitungen mit anschließenden Arbeitsaufträgen, Wiederholungen, Transferaufgaben zu bereits bestehendem Wissen zur Nutzung von Maxima und schließlich aus freiwilligem Einsatz von Maxima. Die konkrete Umsetzung der Idee die Nutzung „integriert“ zu gestalten implizierte einen minimalen, aber sinnstiftenden CAS-Einsatz. Die Aufgaben waren sehr verschieden, trotzdem sollen hier exemplarisch drei Typen vorgestellt werden.

Lassen Sie Maxima die folgenden bestimmten Integrale berechnen und begründen Sie, ob das Ergebnis stimmen kann. Skizzieren Sie auf jeden Fall den Ausschnitt des Graphen per Hand:

$$A = \int_{0,5}^{1,605} \ln(x) dx$$

$$B = \int_0^{\pi} \sqrt{\sin(x)} dx$$

$$C = \int_{-1}^1 |x| dx$$

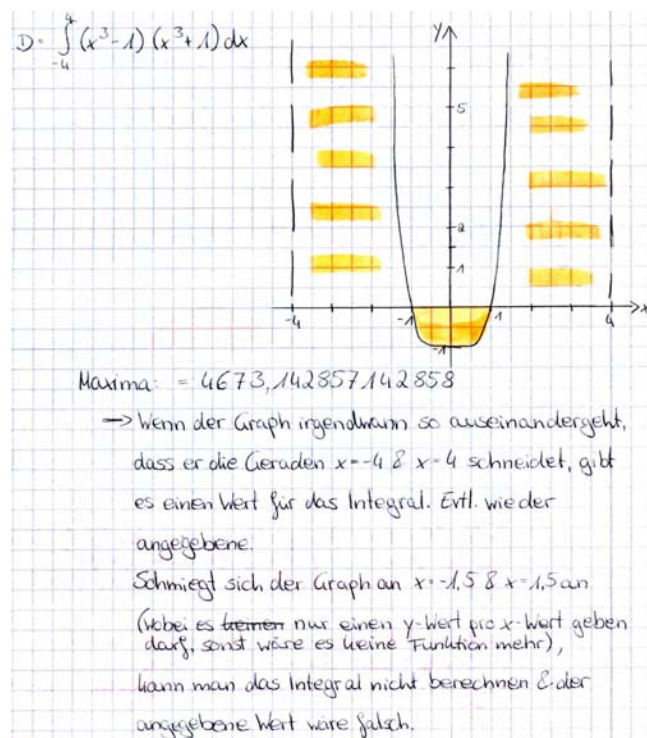
$$D = \int_{-4}^4 (x^3 - 1)(x^3 + 1) dx$$

²Zum Zeitpunkt der Befragung waren dies die einzigen Lehramtsstudierenden, die gerade eine Analysis-Anfängervorlesung hörten.

a) Rechnen lassen und Ergebnisse interpretieren

In dieser Aufgabe wurden bewusst „schwierige“ Integranden gewählt, damit die Hürde des Rechnens per Hand einigermaßen hoch blieb. Eine Anleitung zur Berechnung bestimmter Integrale mit Maxima fand sich auf dem Aufgabenblatt. Im Fokus der Aufgabenstellung steht die Interpretation des Ergebnisses.

Die abgebildete Lösung einer Studentin zeigt beispielhaft, wie die inhaltliche Auseinandersetzung mit dieser Aufgabe ablaufen kann: Der Schwerpunkt der Bearbeitung liegt in der Auseinandersetzung mit dem Ergebnis. Das CAS dient nur zur Berechnung, als Black Box. Der Output muss aber dennoch hinterfragt werden, wie in jedem Modellierungsprozess üblich.



b) Durch Anschauung Erkenntnis gewinnen

Untersuchen Sie, ob die Funktion mit der Gleichung

$$f(x) = \begin{cases} \cos\left(\frac{1}{x}\right) & \text{für } x \neq 0 \\ 0 & \text{für } x = 0 \end{cases}$$

stetig an der Stelle $x = 0$ ist.

Bei dieser Aufgabe handelte es sich um eine Transferaufgabe, da die Variante für $\sin\left(\frac{1}{x}\right)$ bereits in der Vorlesung besprochen worden war. Die Studierenden wussten bereits, wie man Graphen mit Maxima zeichnet, hatten aber hierbei Gelegenheit, selbsttätig Zoomausschnitte zu wählen, um sich das Verhalten des Graphen „um die Null herum“ vor Augen zu führen.

c) Selbst entscheiden, wo und wie CAS eingesetzt wird

Bestimmen Sie zu den folgenden Funktionen jeweils eine Stammfunktion, alle Stammfunktionen (ähnlich Blatt 11, Maxima), das bestimmte Integral über dem Intervall $[1; 2]$ und die Integralfunktion.

Schließlich gab es Aufgaben wie diese, in denen frei gewählt werden konnte, ob Maxima eingesetzt wird oder nicht. Der Verweis zu Blatt 11 bezieht sich auf das Zeichnen von Richtungsfeldern mittels `plotdf`, in denen eine Stammfunktion als Trajektorie dargestellt werden kann.

Ergebnisse

Erwartungsgemäß dauerte es einige Wochen, bis die meisten Studierenden überzeugt waren, dass sie Maxima installieren sollten. Im Vergleich zur Schule ist die Möglichkeit der direkten Einflussnahme an der Universität auf die Studierenden beschränkt.

Etwa 5 Wochen nach Semesterbeginn wurde eine der Übungen so gestaltet, dass die Hausaufgaben selbstständig in Gruppen besprochen werden konnten, so dass die Dozentin im Verlauf der Übung alle Gruppen zur entsprechenden Maxima-Hausaufgabe interviewen konnte. Jede und jeder Studierende wurde also in dieser Übung persönlich gefragt, ob sie/er Maxima schon installiert habe, ob sie/er es schon einmal genutzt habe, und an welchen Stellen noch Hilfe benötigt wird.

Diese offenbar sinnvolle Zwischen-Evaluation zur Nutzung ergab, dass einige Studierende das Programm noch nicht installiert hatten, weil es Probleme dabei gab. Die meisten der Studierenden hatten das Programm zwar installiert, hatten aber noch mehr oder weniger große Hemmungen damit zu arbeiten und nur für einzelne Studierende stellte die Nutzung von Maxima kein Problem dar.

Abgesehen von der technischen Hürde ist die sprachliche Hürde für Anfänger nicht zu unterschätzen. Insbesondere diejenigen, die des Englischen nicht mächtig sind, berichteten von Problemen mit Maxima, da beispielsweise die Maxima-Hilfe auch in der deutschen Version englisch ist. Mit kurzen Anleitungen wurde bei den folgenden Aufgaben versucht, diese Probleme zu beheben. Für den kommenden Zyklus wird eine kurze, auf den Bedarf zugeschnittene Maxima-Anleitung auf Deutsch zur Verfügung stehen.

Befragung der HR-Studierenden zum Maxima-Einsatz

In der letzten Übung mussten die Studierenden einen Fragebogen zum Maxima-Einsatz ausfüllen; wir hatten einen Rücklauf von 24 Fragebögen. Die umseitigen Diagramme zeigen die Ergebnisse zu einigen Fragen. Dabei ist die x -Achse codiert mit 1 für „ja“, 2 für „eher ja“, 3 für „eher nein“ und 4 für „nein“; die y -Achse gibt die Zahl der entsprechenden Antworten an.

Resümee

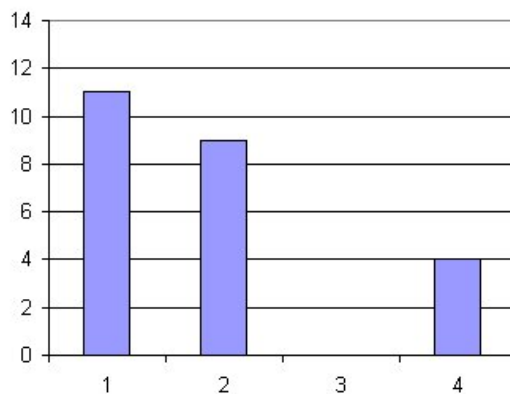
Bevor man allzu unzufrieden mit der Resonanz der Studierenden ist, sollte man berücksichtigen, dass die

überwiegende Mehrzahl der Studierenden diese Vorlesung nicht als direkt nützlich für ihren späteren Beruf als Haupt-, Real- oder Sonderschul-Lehrer empfindet. Dies ist zum Teil verständlich, weshalb wir ja auch in den BA-Studiengängen diese Vorlesung nur als Wahlpflicht-Veranstaltung ausgewiesen haben. Allerdings sollten zumindest die zukünftigen Realschullehrer didaktische Prinzipien wie das Konzept der „Zone der nächsten Entwicklung“ von Vygotskij nicht nur im Zusammenhang mit der Primarstufe sehen, sondern auch verstehen, dass viele ihrer Schülerinnen und Schüler nach dem Real-

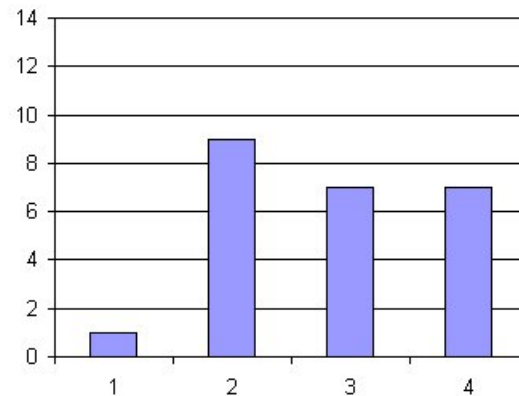
schulabschluss auf ein Fach-Gymnasium wechseln und dort mit Analysis konfrontiert werden. Also sollte ein Realschullehrer selbst wissen, wie es nach der Realschule weitergeht!

Literatur

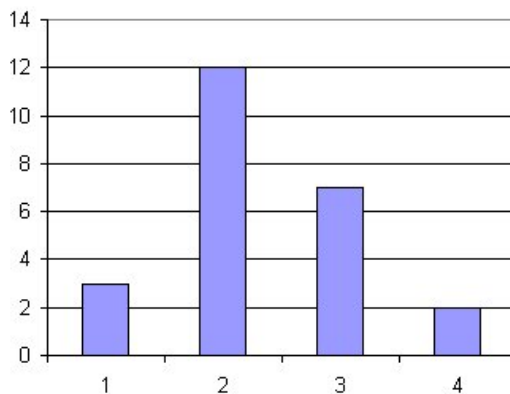
- [1] H.-W. Henn, Lehrerbildung an der Universität Dortmund. Computeralgebra-Rundbrief 38, 2006, S. 16-21.



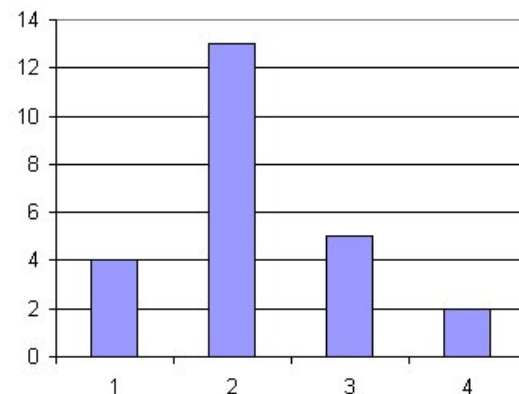
Ich arbeite täglich am Computer.



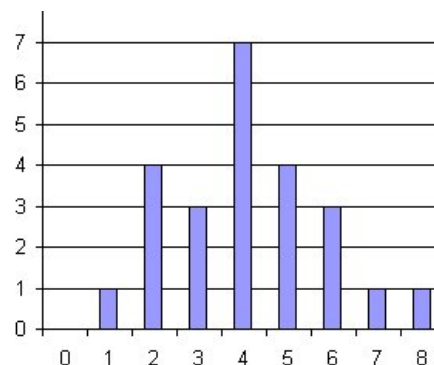
Ich hab schon vor der Elementaren Analysis Mathematikprogramme genutzt.



Ich würde Maxima selbständig in anderen Vorlesungen nutzen, falls es sich anbietet.



Ich habe die Arbeit mit dem Programm Maxima als Bereicherung empfunden.



Anzahl der bearbeiteten Maxima-Aufgaben (von insgesamt acht)

Publikationen über Computeralgebra

- Calmet, J., Ida, T., Wang, D. (Eds.), *Artificial Intelligence and Symbolic Computation*, Lecture Notes in Artificial Intelligence 4120, Springer Verlag, Berlin, Heidelberg, New York, 2008, 689+XX Seiten, ISBN 978-3-540-73541-0, € 53,45.
- Greuel, G.-M., *A Singular Introduction to Commutative Algebra*, Springer Verlag, Berlin, Heidelberg, New York, 2008, 689+XX Seiten, ISBN 978-3-540-73541-0, € 53,45. (Eine Besprechung finden Sie auf Seite 36.)
- Levandovskyy, V., *Non-Commutative Gröbner Bases and Applications*, 2007, Journal of Symbolic Computation (Sonderausgabe), Volume 42 (11-12), 1001-1154. (Eine Besprechung finden Sie auf Seite 37.)
- Wang, D. M., Zhi, L. H. (Eds.), *Symbolic-Numeric Computation*, Birkhäuser Verlag, Basel, Boston, 2007, 384 Seiten, ISBN 978-3-764-37983-4, € 85,49.
- Westermann, T., *Mathematische Probleme lösen mit Maple: Ein Kurzeinstieg*, 3., aktualisierte Auflage, Springer Verlag, Berlin, Heidelberg, New York, 2008, 169 Seiten, ISBN 978-3-540-77720-5, € 23,95. *
- Ziegenbalg, J., O. und B., *Algorithmen. Von Hammurapi bis Gödel*, 2., verbesserte Auflage, Verlag Harri Deutsch Frankfurt am Main, 2007, 374 Seiten, ISBN 978-3-8171-1814-4, € 19,80. (Eine Besprechung finden Sie auf Seite 38.)

* Diese Bücher können auf der Seite <http://www.fachgruppe-computeralgebra.de/Buecher/> oder direkt bei Eva Zerz (eva.zerz@math.rwth-aachen.de) zur Besprechung angefordert werden. Auf der Webseite finden Sie auch noch weitere, hier nicht genannte Bücher zur Besprechung.

Besprechungen zu Büchern der Computeralgebra

G.-M. Greuel, G. Pfister

A Singular Introduction to Commutative Algebra

Springer Verlag, 2008 (2. Auflage), 689+XX Seiten, ISBN 978-3-540-73541-0, € 53,45

Mit dem Computeralgebrasystem SINGULAR kann man Berechnungen in (kommutativen) Polynomringen mit Koeffizienten in gewissen Körpern sowie in deren Faktorringen und Lokalisierungen (in maximalen Idealen) ausführen. Das vorliegende Buch führt in die Kommutative Algebra und damit auch in die Algebraische Geometrie so weit ein, wie das zum Verständnis der in SINGULAR verwendeten Begriffe und Algorithmen nötig ist. Die Auswahl der Themen erfolgte mit Blick auf deren Bedeutung für die Algebraische Geometrie und die Singularitätentheorie. Die Autoren suchen nicht nach größter Allgemeinheit, sondern beschränken sich auf jene Bereiche, die in einem Computeralgebrasystem darstellbar sind. Alle Begriffe, Algorithmen und Konzepte werden im Buch durch mit SINGULAR gerechnete Beispiele verdeutlicht. Beim Lesen hat man das angenehme Gefühl, dass das Buch viel an Inhalt vermittelt, dabei aber immer im positiven Sinn „am Boden bleibt“. Das Fehlen von Hinweisen auf Anwendungen der vorgestellten Algorithmen (zum Beispiel in der ganzzahligen Optimierung, der Systemtheorie oder der Statistik) beeinträchtigt den sehr guten Gesamteindruck nur wenig.

Die ersten zwei Kapitel führen in die algorithmische Theorie der kommutativen Ringe und Moduln ein.

Es werden die Theorie der Gröbnerbasen und der Standardbasen erläutert und deren grundlegende Anwendungen auf Berechnungen mit Idealen und Untermoduln besprochen („Gröbner-basics“). In der vorliegenden zweiten Auflage wurde das erste Kapitel um einen Abschnitt ergänzt: Mit einem Beitrag von Viktor Levandovskyy über Gröbnerbasen in G-Algebren, einer Klasse von gewissen assoziativen Algebren, wird der Erweiterung von SINGULAR auf das Rechnen in nichtkommutativen Ringen Rechnung getragen. Wichtige Beispiele für G-Algebren sind Weyl-Algebren und universelle Einhüllende von endlichdimensionalen Lie-Algebren.

Im Kapitel 3 wird die Noethersche Normalisierung einer endlich erzeugten Algebra besprochen und ein auf einem Normalitätskriterium von Grauert und Remmert fußender Algorithmus zu ihrer Berechnung angegeben. Im Kapitel 4 geht es um die Berechnung der Primärzerlegung und des Radikals eines Ideals. Neu gegenüber der ersten Auflage sind Abschnitte über charakteristische Mengen (für die Berechnung der minimalen assoziierten Primideale) und über Dreiecksmengen (für ein Verfahren zur Lösung von nulldimensionalen Systemen polynomialer Gleichungen). Kapitel 5 ist dem Hilbert-Polynom bzw. dem Hilbert-Samuel-Polynom und deren Anwendung auf die Dimensions-

theorie gewidmet. Kapitel 6 handelt von formalen Potenzreihen, dem Weierstraßschen Vorbereitungssatz und der Berechnung von Standardbasen von Idealen in Potenzreihenringen. Das abschließende Kapitel 7 ist eine Einführung in die (algorithmische) homologische Algebra.

Das letzte Drittel des Buches besteht aus drei Anhängen: Anhang A liefert den geometrischen Hintergrund für die im Hauptteil des Buches eingeführten Begriffe und Resultate der kommutativen Algebra. Die-

ser Abschnitt ist eine sehr konkrete und anschauliche Einführung in die algebraische Geometrie. Anhang B (neu in der zweiten Auflage) beschreibt Algorithmen zur Faktorisierung von Polynomen in einer und mehreren Variablen über endlichen Körpern, \mathbb{Q} und algebraischen Erweiterungen davon. Anhang C führt in die Benutzung des Computeralgebrasystems SINGULAR ein, dieser Anhang wurde in der zweiten Auflage im Hinblick auf die Version 3-0-3 überarbeitet.

Franz Pauer (Innsbruck)

V. Levandovskyy (Ed.) Non-Commutative Gröbner Bases and Applications

Journal of Symbolic Computation (Sonderausgabe), Volume 42 (11-12), 1001-1154, 2007

Im Rahmen des *Special Semester on Gröbner Bases and Related Methods*, das im Frühsommer 2006 in Linz von RISC und RICAM organisiert wurde, fand auch ein Workshop mit dem Titel *Gröbner Bases in Symbolic Analysis* statt. Ein Teil dieses Workshops beschäftigte sich mit nicht-kommutativen Gröbnerbasen und wurde von Viktor Levandovskyy organisiert. Er legt nun auch mit diesem Sonderheft des Journal of Symbolic Computation Proceedings für diesen Teil des Workshops vor (Proceedings für einen anderen Teil dieses Workshops sind als Buch in der RICAM-Reihe bei Walter de Gruyter erschienen und werden im nächsten Beitrag besprochen).

Die Arbeiten decken ein weites Spektrum an Themen ab. Auffällig ist, daß sich kaum ein Beitrag direkt mit der Theorie von Gröbnerbasen beschäftigt, sondern das Schwergewicht auf Anwendungen in verschiedenen Situationen liegt. Dementsprechend werden auch kaum Implementierungsfragen diskutiert; alle Beiträge sind eher theoretisch orientiert (Viktor Levandovskyy weist allerdings in seinem Vorwort auf eine Datenbank unter [http://www.risc.uni-linz.ac.](http://www.risc.uni-linz.ac.at/Groebner-Bases-Implementations)

[at/Groebner-Bases-Implementations](http://www.risc.uni-linz.ac.at/Groebner-Bases-Implementations), die im Rahmen des Workshops auch um eine Reihe nicht-kommutativer Einträge ergänzt wurde). Mit Ausnahme eines Übersichtsartikel von Pauer zu Gröbnerbasen über Noetherschen Ringen enthält das Heft nur Originalarbeiten.

Im Einzelnen sind folgende Arbeiten in dem Band enthalten: F. Pauer, *Gröbner bases with coefficients in rings*; E.L. Green, Ø. Solberg, *An algorithmic approach to resolutions*; G.A. Evans, C.D. Wensley, *Complete involutive rewriting systems*; J.M. Casas, M.A. Insua, M. Ladra, *Poincaré-Birkhoff-Witt theorem for Leibniz n -algebras*; P. Cameron, N. Iyudu, *Graphs of relations and Hilbert series*; T. Gateva-Ivanova, S. Majid, *Set-theoretic solutions of the Yang-Baxter equation, graphs and computations*; A. Quadrat, D. Robertz, *Computation of bases of free modules over the Weyl algebras*; R.M. Falcón, J. Martín-Morales, *Gröbner bases and the number of Latin squares related to autotopisms of order ≤ 7* .

Werner M. Seiler (Kassel)

M. Rosenkranz, D. Wang (Eds.) Gröbner Bases in Symbolic Analysis

Radon Series in Computational and Applied Mathematics 2, Walter de Gruyter, Berlin, New York 2007, ISBN 978-3-11-0193237

Der vorliegende Band gibt elf Übersichtsvorträge wieder, die teilweise auf dem Workshop mit demselben Titel vom 08. bis 17.05.2006 in Linz im Rahmen des *Special Semester on Gröbner Bases and Related Methods* vom Radon Institute for Computational and Applied Mathematics (RICAM) und dem Research Institute for Symbolic Computation (RISC) gehalten wurden.

Breit gefächert wie die Themen von theoretischen, algorithmischen bis zu anwendungsorientierten Aspekten sind die wissenschaftlichen Ausgangspositionen und Hintergründe der Autoren, von Physik und Ingenieurwissenschaften über Analysis, Differentialgleichungen und Numerik, kommutativer Algebra und Differentialalgebra bis hin zur Informatik. Es ist erstaunlich, wen

die Gröbnerbasen alle unter einem Dach vereinigen. Die Herausgeber nennen als Schwerpunkte Algebraische Analysis, Involutive Basen und Janetbasen, Differentialalgebra, Lie-Gruppen und Invariantentheorie, Rand- und Anfangswertaufgaben. Die Grenzen sind fließend, die Sprache nicht immer einheitlich, jedoch mit starken Vereinheitlichungstendenzen, so dass die Zeit langsam für ein grundlegendes Lehr- und Referenzbuch für das neu entstehende Gebiet reif zu werden scheint.

Man kann diesen Band als einen wichtigen Teilschritt in diese Richtung sehen, sind doch insbesondere die durchweg ausführlichen Literaturangaben von großem Wert. Hier eine Auflistung der einzelnen Arbeiten mit Kurzkomentaren zum Inhalt, soweit er über eine allgemeine Vorstellung des Gebietes hinausgeht:

J. F. Pommaret, *Gröbner Bases in Algebraic Analysis: New Perspectives for Applications*, Anwendungen in der Kontrolltheorie unter neuer Einbringung bekannter algebraischer Konzepte (purity);

U. Oberst, F. Pauer, *Solving Systems of Linear Partial Difference and Differential Equations with Constant Coefficients Using Gröbner Bases*, Überblick algorithmischer und grundlegender Aspekte der in der Arbeitsgruppe entwickelten Theorie mit historischen Kommentaren (Riquier);

A. Levin, *Computation of the Strength of Systems of Difference Equations via Generalized Gröbner Bases*, Übertragung theoretischer Konzepte der Differentialalgebra auf den Differenzenfall im Hinblick auf die Entwicklung algorithmischer Methoden;

G. Carra Ferro, *A Survey on Differential Gröbner Bases*, grundlegendes Begriffsgebäude und Algorithmen, insbesondere im Hinblick auf Existenz der Differentialgröbnerbasen;

F. Boulier, *Differential Elimination and Biological*

Modelling, Vorstellung der Pakete diffalg und BLAD für Zwecke der Differentialelimination mit biologischen Anwendungen;

D. Robertz, *Janet Bases and Applications*, Vorstellung involutiver Basen für diverse Kontexte inklusive Ore-Algebren mit zugehörigen Paketen Involutive, Janet, JanetOre und Anwendungsbeispielen in Gruppentheorie und Kontrolltheorie;

W. M. Seiler, *Spencer Cohomology, Differential Equations, and Pommaret Bases*, Homologische Konzepte im Umfeld des Satzes von Cartan-Kähler, theoretische Kommentare zu Involutiven Basen in diesem Kontext, Gegenüberstellung Castelnuovo-Mumford-Regularität und Involutivität;

P. Olver, J. Pohjanpelto, *Differential Invariants for Lie Pseudo-Groups*, Anwendung der Cartanschen Moving-Frame-Methode auf Äquivalenzprobleme mit Erzeugung von Differentialinvarianten;

W. N. Traves, *Invariant Theory and Differential Operators*, Überblick der konstruktiven Invariantentheorie mit motivierenden Beispielen insbesondere bei Invarianten in der Weylalgebra;

K. Krupchyk, J. Tuomela, *Compatibility Complexes for Overdetermined Boundary Problems*, Homologische Konzepte anwendungsnaher Differentialgleichungssysteme mit Preprocessing für numerische Rechnungen;

F. L. Pritchard, W. Y. Sit, *On Initial Value Problems for Ordinary Differential-Algebraic Equations*, Überblick bekannter und Vorstellung neuer formaler Methoden für Vorbehandlung numerischer Probleme.

Jeder, der sich für formale Methoden für Differentialgleichungen interessiert, wird dieses Buch anregend finden.

Wilhelm Plesken (Aachen)

B., J. und O. Ziegenbalg Algorithmen von Hammurapi bis Gödel

Verlag Harri Deutsch, 2007, 374 Seiten, ISBN 978-3-8171-1814-4, € 19,80

Von der Auffassung ausgehend, dass Algorithmen im Zentrum unserer Wissenschafts- und Kulturgeschichte stehen, schlägt dieses Buch einen weiten Bogen. Es beginnt mit der Diskussion, was man unter Informatik und Algorithmen zu verstehen habe. Eine elementare Einführung in historische Bezüge anhand einiger alter Verfahren schließt sich an. Der Hauptteil besteht aus der Vorstellung unterschiedlichster Strategien zum algorithmischen Problemlösen und der Untersuchung der Effizienz von Algorithmen. Ein längeres Kapitel über Programmierstile und Programmiersprachen behandelt die Sprachen Pascal, Lisp, Logo, Scheme, Prolog, BASIC

und Prolog. Den Epilog bildet ein Exkurs in genetische Algorithmen und neuronale Netze.

Viele Beispiele werden behandelt, wobei die meisten Algorithmen in Pascal und/oder Mathematica (in einer Version < 6.0) implementiert werden. Das Niveau dieser Beispiele ist sehr unterschiedlich, so dass unklar bleibt, an wen sich dieses Buch wendet.

Es handelt sich um die überarbeitete Auflage eines älteren Buches; dies merkt man, wenn im Jahr 2007 immer noch Pfennige, MS-DOS, OS/2 o. Ä. auftauchen, eine Sprache wie Java jedoch nicht.

Elkedagmar Heinrich (Konstanz)

1. Jahrestagung der GDM 2008

Budapest, Ungarn, 13. – 20.03.2008

<http://mathdid.elte.hu/GDM2008/gdm2008.html>

Knapp 400 Mathematikdidaktiker trafen sich zur 42. Jahrestagung der Gesellschaft für Didaktik der Mathematik in diesem Jahr in Budapest – ein interessantes Novum! Neben den sechs Hauptvorträgen waren Sektionsvorträge und selbstmoderierte Sektionen, darunter etliche zum Thema Computeralgebra, der Kern der Tagung. Eine Posterausstellung und die Sitzungen der GDM-Arbeitskreise komplettierten das wissenschaftliche Programm. Der Ausflugstag und die vielfältigen Angebote von Kaffeepausen bis zu gemeinsamen Abendessen sorgten dafür, dass alte Freundschaften gefestigt und neue gefunden werden konnten und dass die schöne Stadt Budapest zu ihrem Recht kam.

Hans-Wolfgang Henn (Dortmund)

2. Computeralgebra in Lehre, Ausbildung und Weiterbildung VI: Computeralgebra und ihre Didaktik – Einfluss auf Lernen und Prüfen

Landesinstitut für Schule/Qualitätsagentur Soest, 27. – 28.03.2008

<http://www.fachgruppe-computeralgebra.de/CLAW>

Einen ausführlichen Bericht zu dieser Tagung finden Sie auf Seite 7 in diesem Rundbrief.

3. SCC 2008 – First International Conference on Symbolic Computation and Cryptography

Peking, China, 28. – 30.04.2008

<http://www.cc4cm.org/scc2008>

SCC 2008 war die erste in einer geplanten Reihe von Konferenzen über neue Entwicklungen im Schnittbereich der Computeralgebra mit der Kryptographie. Es gelang den Organisatoren unter der Leitung von Zhiming Zheng, drei Hauptvortragende von höchstem Ansehen zu gewinnen: Bruno Buchberger (Linz) sprach über *Gröbner Bases in Theorema using Functors*, Adi Shamir (Weizmann Institut, Israel) gab einen Überblick über *Algebraic Techniques in Cryptography* und Frau Xiaoyun Wang (Peking) erklärte ihre *RSA Cryptanalysis Based on the LLL Algorithm*.

Dazu kamen ca. 20 weitere Vorträge über aktuelle Versuche, mit Hilfe von Techniken der Computeralgebra Kryptosysteme zu knacken, oder neue Vorschläge für Kryptosysteme, die auf Methoden der Computeralgebra beruhen. Besondere Beachtung fanden dabei die Beziehungen zwischen Gröbnerbasen in nicht-kommutativen Ringen und Kryptosystemen, die nicht-kommutative algebraische Strukturen (z. B. Gruppen) verwenden, sowie Angriffe auf vereinfachte Versionen des aktuellen eStream-Finalisten Trivium.

Ausgewählte Beiträge dieser Tagung werden in einem speziellen Heft der Zeitschrift *Mathematics in Computer Science* im Birkhäuser Verlag erscheinen.

Martin Kreuzer (Passau)

4. Rhine Workshop on Computer Algebra

Levico Terme, Italien, 16. – 20.06.2008

<http://www.science.unitn.it/~degraaf/rwca.html>

Das Grand Hotel Bellavista in Levico Terme (nahe Trient), einem bevorzugten Erholungsort der Habsburger, bot einen prächtigen Veranstaltungsort für diese sehr vielfältige und abwechslungsreiche Konferenz. Besondere Schwerpunkte mit jeweils drei oder mehr der insgesamt 19 Vorträge waren Anwendungen der Computeralgebra in der Gruppentheorie, in Lie-Algebren, in der Invariantentheorie, und in der algebraischen Geometrie (insbesondere für endliche Punktmen-gen bzw. nulldimensionale Ideale). Obwohl das Wetter nicht immer mitspielte, gelang es in einer Regenspauze, einen Konferenzausflug in Form einer Wanderung auf einem nahegelegenen Berg zu organisieren, und an den Abenden wurden der Beamer und die Leinwand im Konferenzsaal für die Übertragung der Spiele der Fußball-Europameisterschaft genutzt. Trotz oder gerade wegen dieser Ablenkungen herrschte eine sehr rege und diskussionsfreudige Atmosphäre unter dem bunt gestreuten Teilnehmerkreis, der von Georgien über die Türkei bis nach Marokko reichte.

Martin Kreuzer (Passau)

5. ICME 11 – The International Congress on Mathematical Education

Monterrey, Mexiko, 06. – 13.07.2008

<http://extra.shu.ac.uk/iowme/icmi.html>

Der in vierjährigem Turnus stattfindende International Congress on Mathematical Education wurde in diesem Jahr in der nordmexikanischen Millionenstadt Monterrey abgehalten. Etwa 2000 Teilnehmer aus der ganzen Welt, darunter circa 40 aus Deutschland, diskutierten alle Belange der mathematischen Ausbildung – von der Grundschule bis zur beruflichen Weiterqualifizierung. Das wissenschaftliche Programm gliederte sich in bewährter Weise zunächst in neun Plenary Lectures und in fünf einstündige Timeslots mit je ca. zwölf Regular Lectures. Kern der ICME-Arbeit sind die Topic Study Groups und die Discussion Groups, in denen Fachleute aus der ganzen Welt zu speziellen Themen in Kurzreferaten und Diskussionen etwa sechs Stunden im Laufe der Kongressdauer zusammen arbeiten. Weitere Aktivitäten wie Poster Exhibition, Vorstellungen der laufenden ICME-Studies, der ICMI Affiliated Study Groups, einer großen Exhibition usw. vervollständigten das Programm. Bei all diesen Aktivitäten waren auch Beiträge zum Thema Computeralgebra vertreten. Die abendlichen Happy Hours und die Ausflüge sorgten bestens dafür, dass Kolleginnen und Kollegen aus der ganzen Welt alte Kontakte erneuern und neue Kontakte schließen konnten.

Hans-Wolfgang Henn (Dortmund)

6. ISSAC 2008

RISC Linz, Österreich, 20. – 23.07.2008

<http://www.risc.uni-linz.ac.at/about/conferences/issac2008/>

Auf der diesjährigen Tagung der ISSAC, die vom 20.07. bis zum 23.07. im *Research Institute of Symbolic Computation (RISC)*, einem Institut der *Johannes Kepler Universität Linz*, in Linz/Hagenberg stattfand, nahmen 171 Personen teil. Das RISC liegt etwa 20 km von Linz entfernt und wurde im und rund um das alte Schloss Hagenberg errichtet. Die Konferenz selbst fand im Softwarepark Hagenberg, in den das RISC eingebettet ist, statt. Trotz abgelegener Lage war die Infrastruktur hervorragend und die ISSAC 2008 eine Konferenz der kurzen Wege – selbst das Hotel *Sommerhaus*, das zur FH gehörige Studentenwohnheim, war nur fünf Minuten vom Tagungsort entfernt.



Tagungszentrum im Softwarepark Hagenberg

Durch die Einbindung der ISSAC 2008 in den *RISC Summer* fanden vor und nach der Konferenz noch andere Tagungen, Workshops u. Ä. zu verwandten Themen statt, so dass viele Teilnehmer gleich für mehrere Veranstaltungen blieben.

Wie jedes Jahr wurde bei über 100 eingereichten Artikeln von 40 Vortragenden über neue Forschungsergebnisse in der Computeralgebra berichtet. Zusätzlich zum dreitägigen Programm wurden am Sonntag auch wieder zwei Tutorials angeboten. Ferner gab es neben Vorträgen zu gewohnten Themengebieten wie Differentialgleichungen, Gröbnerbasen und algebraischer Geometrie auch einen Vortragsblock mit zahlreichen Softwarepräsentationen. Zwischen den halbstündigen Vorträgen waren drei einstündige Hauptvorträge angesetzt, für die Elizabeth L. Mansfield, Sergei Abramov und William Stein gewonnen werden konnten. Letzterer hielt einen sehr interessanten Vortrag über das Open-Source-Computeralgebrasystem *Sage*.

Beim ISSAC Business Meeting am Montag wurde u. A. über den Austragungsort der ISSAC 2010 abgestimmt. Dabei erhielt München 61 Stimmen und Boston 39 Stimmen, so dass die ISSAC 2010 nach 1998 in Rostock wieder in Deutschland stattfinden wird. Der Gastgeber der ISSAC im nächsten Jahr vom 28.07.09 bis 31.07.09 ist Seoul in Südkorea. Ferner wurde Michael Monagan als Nachfolger des Vorsitzenden des ISSAC-Organisationskomitees Jeremy Johnson gewählt. Er setzte sich knapp gegen seine Konkurrenten David Jeffrey und Agnes Szanto durch.

Am Dienstagabend fand das alljährliche Bankett in festlichem Ambiente bei einer Schiffsrundfahrt auf der Donau statt. Dort wurden auch die diesjährigen Preise der ISSAC und ACM/SIGSAM vergeben.



Bankett auf der Donau

Die Preisträger im Einzelnen:

Preis	dotiert	Preisträger
Ehrenpreis	5000 \$	Bruno Buchberger
Richard D. Jenks-Preis	1000 \$	GAP-Team
Bester Artikel	1000 \$	Adam Strzebonski
Bester Nachwuchsartikel	500 \$	Adrien Poteaux
Bestes Poster	–	Thomas Wolf



Links: Jürgen Gerhard von Maplesoft überreicht Adam Strzebonski von Wolfram Research den Preis des besten wissenschaftlichen Artikels und die neueste Version des Computeralgebrasystems Maple

Rechts: Träger des Ehrenpreises 2008: Bruno Buchberger



Träger des Posterpreises 2008: Thomas Wolf

Zum Abschluss der Tagung wurde am Mittwochabend ein Stadtrundgang in Linz angeboten, in dessen Rahmen die Stadt und ihre Geschichte in Kleingruppen unter kompetenter Führung entdeckt werden konnten.

Auf der Homepage der Tagung (siehe oben) finden Sie weitere Informationen, wie z. B. das komplette Tagungsprogramm, Fotos der Vortragenden und Organisatoren und Links zu früheren ISSAC-Tagungen. Unter der neuen Internetpräsenz <http://www.issac-conference.org> wird in Kürze alles zu vergangenen und kommenden ISSAC-Tagungen zu finden sein.

Peter Horn, Torsten Sprenger (Kassel)

7. ApCoA 2008 – Workshop on Approximate Commutative Algebra

RISC Linz, Österreich, 24. – 26.07.2008

<http://cocoa.dima.unige.it/conference/apcoa2008/>

Dieser Workshop war die zweite Ausgabe der ApCoA-Reihe und setzte den Workshop am RICAM-Institut der Universität Linz vom Februar 2006 fort. Trotz einer (wohl auch auf Grund der massiven Konferenzballung des *RISC Summer 2008*) nicht so großen Teilnehmerzahl von 25 zeigt bereits die Liste der fünf Hauptvortragenden die Exzellenz der Tagung: Hans J. Stetter (Wien) sprach über *Embedding Commutative Algebra into Analysis*, Boris Shekhtman (Tampa, USA) über *Ideal Interpolation*, Zhonggang Zheng (NE Illinois, USA) über *Regularization and Matrix Computation in Numerical Polynomial Algebra*, Andrew J. Sommese (Notre Dame, USA) über *Numerical Decomposition of the Rank-Deficiency Set of a Matrix of Multivariate Polynomials* und Erich Kaltofen (North Carolina, USA) über *Exact Certification in a Global Polynomial Optimization*.

Weitere Beiträge zum Thema des Workshops – den Beziehungen zwischen exakten symbolischen Berechnungen und numerischen Methoden – waren drei verschiedene numerische Varianten des Buchberger-Möller-Algorithmus von Maria-Laura Torrente (Pisa), Claudia Fassino (Genua) und dem Berichterstatter sowie Anwendungen in der symbolisch-numerischen Behandlung von Differentialgleichungen durch Robin Scott und Wenyuan Wu (beide London, Ontario, Kanada). Ein spezieller Band der RISC-Buchserie (Springer Verlag) mit den Proceedings des Workshops ist in Vorbereitung.

Martin Kreuzer (Passau)

8. ACA 2008 – Applications of Computer Algebra

RISC Linz, Österreich, 27. – 30.07.2008

<http://www.risc.uni-linz.ac.at/about/conferences/aca2008/>

Die Konferenz „Applications of Computer Algebra (ACA)“ fand im Zeitraum 27.–30.07.2008 am Research Institute for Symbolic Computation (RISC) in Hagenberg bei Linz, Österreich, statt. Die Tagung war Teil der Veranstaltungsreihe *RISC Summer 2008* und hatte 192 registrierte Teilnehmer. Es gab 15 einzelne Sektionen zu verschiedensten Themen wie z. B. *Symbolic and Algebraic Computation for Optimization Tasks in Science and Engineering*, *Symbolic and Numeric Computation*, *Algebraic and Algorithmic Aspects of Differential and Integral Operators*, *Symbolic Computation and Quantum Field Theory*, *Gröbner Bases and their Applications* usw. Neben den 30-minütigen Sektionsvorträgen, von denen jeweils vier parallel stattfanden, gab es zwei einstündige Plenarvorträge: einen von A. Akritas zum Thema *Computing bounds on the values of positive roots of polynomials* und einen von S. Watt mit dem Titel *How to work with polynomials of symbolic degree*. In letzterem Vortrag wurde u. A. der Unterschied zwischen Computeralgebra und allgemeinem symbolischen Rechnen diskutiert. Höhepunkt war ein wunderbares „Dinner at the castle“ im Schloss Hagenberg mit einem mehrstündigen Auftritt des „Dental Jazz Trio“, in dem Prof. Dr. Bruno Buchberger die Klarinette spielte. Das Bankett fand im Linzer Stadtkern statt und bot zahlreiche Überraschungen und Preise für die Teilnehmenden. Die nächste ACA-Tagung wird im Zeitraum 25.–28.06.2009 in Montreal (Kanada) stattfinden.

Victor Levandovskyy (Aachen)

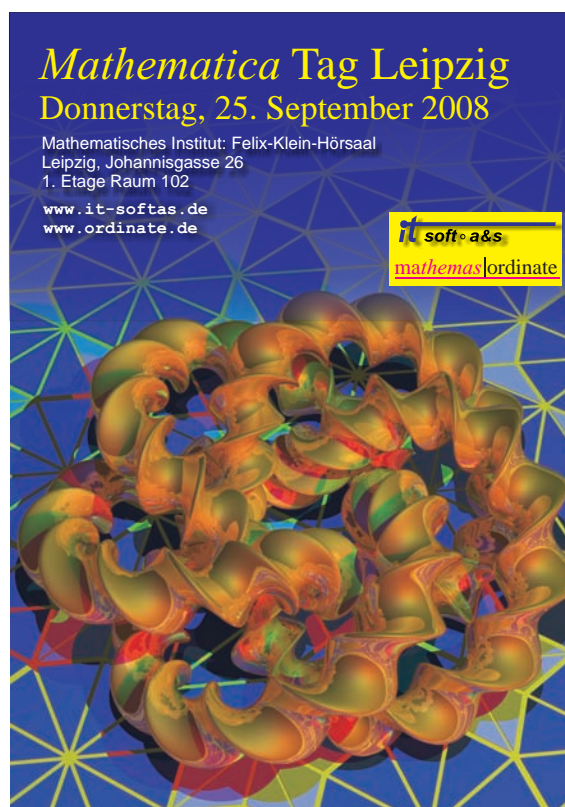
9. Mathematica-Tag

Leipzig, 25.09.2008

<http://www.ordinate.de/mathematicaTag.htm>

Auf diesem von Carsten Herrmann (*mathemas ordinate*) und Dr. Hans-Jörg Möhring (*it soft a & s*) organisierten regionalen Mathematica-Anwendertreffen wurden in drei Vorträgen Aspekte des Mathematica-Einsatzes besprochen. Im Beitrag *Programme optimieren* von Carsten Herrmann ging es um effiziente Anwendungen des funktionalen Paradigmas in einer stark prozedural vorgeprägten Welt. Dr. Jens Kuska (IZBI, Universität Leipzig) stellte in seinem Beitrag *Grafik in Mathematica – MathGL3D* Möglichkeiten vor, wie mit Linienintegral-Faltungen aussagefähige Visualisierungen von Vektorfeldern erzeugt werden können. Diese sind Teil des MathGL3D-Pakets, <http://www.wolfram.com/products/applications/mathgl3d>. Ein Highlight bot der Beitrag *Mathematica 6 – New Directions in Technical Computing* von Jon McLoone (Senior Developer, Wolfram Research Europe), in dem wichtige Entwicklungslinien von Mathematica hin zu einem komplexen Werkzeug des wissenschaftlichen Arbeitens aufgezeigt wurden. Der Schwerpunkt der Ausführungen lag auf den integrativen Fähigkeiten, die sich aus der Verbindung von symbolischen, numerischen und grafischen Aspekten in hybriden Verfahren, aus der Online-Verfügbarkeit gepflegter Datenbestände (Wetterdaten, Finanzdaten, Human-Genom-Projekt usw.) und schließlich den Parallelisierungsmöglichkeiten im persönlichen Grid auf dem Weg zum Netz als Computer ergeben.

Hans-Gert Gräbe (Leipzig)



1. Elfter Mitteldeutscher Computeralgebra-Tag und T^3 -Regionaltagung Mitteldeutschland

Halle, 10.10.2008

<http://www.informatik.uni-leipzig.de/~graebe/MCAT/mcat11.html>

Es ergeht die herzliche Einladung zum Elften Mitteldeutschen Computeralgebra-Tag. Als Vortragende haben zugesagt: Peter Schenzel (Halle), Heinz Klaus Strick (Leverkusen), Wolfgang Moldenhauer (Bad Berka), Ines Petschler (Leipzig). Neben den Vorträgen wird es zwei Arbeitsgruppen geben:

1. *Dynamische Geometrie* (Ines Petschler, Leipzig): Aufbauend auf den Vortrag können die Teilnehmenden selbst mit Geogebra arbeiten und erste Erfahrungen sammeln. Es werden verschiedene Aufgaben angeboten, so dass auch diejenigen, die bereits mit Geogebra gearbeitet haben, herzlich eingeladen sind.

2. *Stochastik in der Kursstufe mit CAS-Rechnern* (Elvira Malitte, Universität Halle-Wittenberg): Die Teilnehmenden können eigene Erfahrungen beim Einsatz von CAS-Rechnern für die Stochastik der Kursstufe sammeln, Chancen und Grenzen diskutieren. Die Rechner werden zur Verfügung gestellt. Vorkenntnisse im Umgang mit den Rechnern werden nicht vorausgesetzt.

Organisation:

Elvira Malitte und Peter Schenzel (Halle), Hans-Gert Gräbe (Leipzig)

2. ACAT 2008 – International Workshop on Advanced Computing and Analysis Techniques in Physics Research

Erice, Italien, 03. – 07.11.2008

<http://acat2008.cern.ch/>

We are very happy to invite you to this exceptional session of the ACAT series (12th) that will mark a new turning point in the cross-fertilization of hot physics research and computing technology.

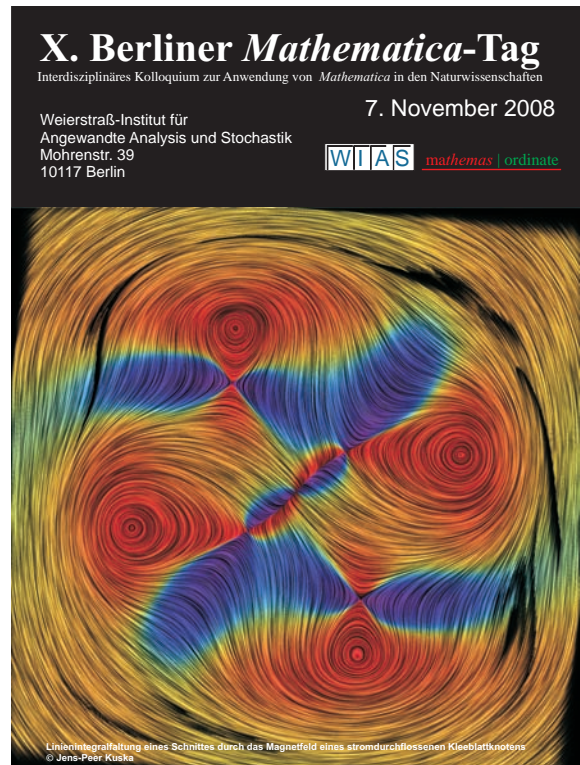
The ACAT workshop series, created back in 1990 as AI-HENP (Artificial Intelligence in High Energy and Nuclear Research) has been covering the tremendous evolution of computing in its most advanced topics, trying to setup bridges between computer science, experimental and theoretical physics.

3. Zehnter Berliner Mathematica-Tag

Berlin, 07.11.2008

<http://www.ordinate.de/mathematicaTag.htm>

Der zehnte Berliner Mathematica-Tag wird am 07.11.2008 stattfinden – ein interdisziplinäres Kolloquium zur Anwendung von Mathematica in den Naturwissenschaften. Gastgeber ist wie in den Jahren zuvor das WIAS Berlin. Nähere Informationen sind demnächst auf der oben genannten Webseite verfügbar.



4. Mathematical Methods in Computer Science 2008

Karlsruhe, 17. – 19.12.2008

<http://iaks-www.ira.uka.de/eiss/index.php?id=28>

The Mathematical Methods in Computer Science conference will be held in memory of Thomas Beth. Its scope is defined by the research areas of Thomas Beth. Topics of interest include, but are not restricted to coding theory, cryptography and symbolic computation.

5. 80. Jahrestagung der GAMM

Danzig, 09. – 13.02.2009

<http://www.gamm2009.pl>

The GAMM Annual Meeting 2009 will be held in the main campus of Gdansk University of Technology, which is the oldest and the biggest one in the North of Poland.

The sessions will take place in the ETI New Auditorium (ETI NA), the Main Building (MB), the Chemical Faculty Buildings (ChB) and Auditorium Novum (AN). All buildings are connected by the university foot-paths.

6. Jahrestagung der GDM 2009

Oldenburg, 02. – 06.03.2009

<http://didaktik-der-mathematik.de/>

Die Gesellschaft für Didaktik der Mathematik (GDM) lädt herzlich zu ihrer Jahrestagung 2009 nach Oldenburg ein.

Nähere Informationen sind demnächst auf der oben genannten Webseite verfügbar.

7. GCR 2009 – Geometric Constraints and Reasoning

Honolulu, Hawaii, 08. – 12.03.2009

<http://www.lsi.upc.edu/~robert/gcr2009/gcr2009.html>

Geometric Constraints and Reasoning (GCR) is a technical track of the International Symposium on Applied Computing. For the past twenty years, the ACM Symposium on Applied Computing (SAC) has been a primary forum for applied computer scientists, computer engineers and application developers to gather, interact, and present their work.

SAC is sponsored by the ACM Special Interest Group on Applied Computing (SIGAPP), its proceedings are published by ACM in both printed form and CD-ROM; they are also available on the web through ACM's Digital Library.

As a special track of SAC, GCR is devoted to geometric reasoning taken in a broad sense. Initially, this track focused on geometric constraint solving but it appears that geometric computing and reasoning is closely related to this topic. Our aim is then to widen the audience and to make GCR a place where the communities of geometric constraint solving, computer aided deduction in geometry and related disciplines can meet and have fruitful exchanges.

SAC 2009 is also an opportunity to attend tracks related to GCR, about combinatorial optimization, constraint programming (non geometrical constraints), graph algorithms, numerical methods or interval analysis, etc.

8. CASK – Computeralgebra-Symposium Konstanz

Konstanz, 12. – 13.03.2009

<http://www.cask.htwg-konstanz.de>

Wir laden herzlich ein zum Computeralgebra-Symposium in Konstanz am Bodensee.

Zunehmend bringen Studenten CA-Erfahrung von der Schule mit und sind entsetzt, wenn sie an der Hochschule auf einmal wieder mit Papier und Bleistift, aber ohne Rechneinsatz, mathematische Aufgaben bearbeiten sollen – eine Tätigkeit, die sie die letzten Schuljahre hindurch nicht mehr geübt haben. Lehrende, die dies fordern, wenden sich gegen die Mentalität des „Knöpfeledrückers“ (Originalzitat), der gar nicht mehr wisse, was er tut, und erst einmal mathematische Fertigkeiten lernen müsse.

Dagegen steht die Meinung derer, die fragen, was denn der grundsätzliche Unterschied zwischen der Benutzung eines Rechenschiebers oder einer Logarithmentafel einerseits und dem Einsatz eines Taschenrechners oder CAS andererseits sei; schließlich müsse man immer wissen, was man berechnen will und wie man dafür vorzugehen hat.

Gleichzeitig liegen viele Erfahrungen vor, daß durch den CA-Einsatz zwar gute Studenten profitieren, schlechte aber noch schwächere Leistungen zeigen. Wie kann Lehre so gestaltet werden, daß durch die Nutzung von CA alle Studenten profitieren und das mathematische Verständnis steigt? Wie sollte das Curriculum weiterentwickelt werden? Wie müssen Klausuren gestaltet werden, bei denen der Einsatz eines CAS zulässig ist? Lassen sich (in Zeiten von WLAN und anderen technischen Möglichkeiten) Klausuren mit Rechnernutzung überhaupt betrugssicher organisieren?

Neben der Diskussion über diese und ähnliche Fragen sollen Berichte aus der Forschung die vielseitigen Aspekte der Anwendung von CA illustrieren.

Ort der Tagung ist der Campus der HTWG Konstanz, etwa 1,5 km vom Hauptbahnhof Konstanz entfernt.

Organisation:

E. Heinrich, H.-D. Janetzko

9. Tagung der Fachgruppe Computeralgebra

Kassel, 14. – 16.05.2009

<http://www.fachgruppe-computeralgebra.de>

Diese Tagung setzt die Reihe der Tagungen der Fachgruppe (Kassel 2003, 2005, Kaiserslautern 2007) fort.

Über die Hauptvorträge der Tagung werden Sie demnächst auf unserer Webseite informiert.

10. CoCoA 2009 – International School on Computer Algebra

Barcelona, Spanien, 08. – 12.06.2009

<http://cocoa.dima.unige.it/conference/cocoa2009/>

Die CoCoA-Schule richtet sich an Diplomanden und Doktoranden aus der ganzen Welt, die an Themen aus der kommutativen Algebra oder algebraischen Geometrie arbeiten und das Computeralgebrasystem CoCoA einsetzen wollen. Es wird zwei Kurse mit zugehörigen Tutorien geben: Marilina Rossi: *On Castelnuovo Regularity and Related Problems* (Tutorien: Anna Bigatti); Anthony V. Geramita: *Secant Varieties* (Tutorien: Enrico Carlini)

11. Conference on Computational Computer Algebra (zu Ehren von Lorenzo Robbiano anlässlich seines 65. Geburtstags)

Barcelona, Spanien, 12. – 13.06.2009

<http://cocoa.dima.unige.it/conference/robbiano65/>

Die Konferenz findet zwischen der CoCoA-Schule und der MEGA 2009 statt. Alle drei Tagungen verwenden denselben Veranstaltungsort an der Universität Barcelona.

Organisation:

K. Ranestad (Oslo), T. Johnsen (Bergen), G. Floystad (Bergen), S. Smaloe (Trondheim)

12. MEGA 2009 – Effective Methods in Algebraic Geometry

Barcelona, 15. – 19.06.2009

<http://www.imub.ub.es/mega09/>

The Conference MEGA 2009 will be held at the University of Barcelona, from Monday, June 15th to Friday, June 19th, 2009.

MEGA is the acronym for Effective Methods in Algebraic Geometry (and its equivalent in Italian, French, Spanish, German, Russian, etc.), a series of roughly biennial conferences on computational and application aspects of Algebraic Geometry and related topics with very high standards. Previous meetings were held in 1990 (Castiglione, Italy), 1992 (Nice, France), 1994 (Santander, Spain), 1996 (Eindhoven, Netherlands), 1998 (St. Malo, France) 2000 (Bath, United Kingdom), 2003 (Kaiserslautern, Germany), 2005 (Porto Conte, Italy) and 2007 (Strobl, Austria).

Proceedings containing a selection of the papers and invited talks presented at previous Mega conferences have been published by Birkhäuser in the series Progress in Mathematics (volumes no. 94, 109 and 143), by the Journal of Pure and Applied Algebra (volumes no. 117 and 118, 139 and 164) and by the Journal of Symbolic Computation (volumes no. 39 3-4 and 42 1-2).

Conference topics include effective methods, complexity issues and applications in: Commutative Algebra, Algebraic and Semialgebraic Geometry, Algebraic Number Theory, Algebraic Geometry and related fields: Algebraic Analysis of Differential Equations, Differential Geometry, Associative Algebras, Group Theory, Algebraic Groups and Lie Algebras, Algebraic and Differential Topology.

13. Summer School on Computer Algebra and Syzygies

Nordfjordeid, Norwegen, 15. – 19.06.2009

<http://www.math.uio.no/div/nordfjordeid/nordfjord.html>

Diese Sommerschule richtet sich an Diplomanden und Doktoranden mit Kenntnissen in Algebra oder algebraischer Geometrie. Es werden maximal 75 Teilnehmer zugelassen. Es finden drei Kurse mit jeweils acht Vorträgen und zugehörigen Übungen statt.

Organisation:

K. Ranestad (Oslo), T. Johnsen (Bergen), G. Floystad (Bergen), S. Smaløe (Trondheim)

14. ACA 2009 – 15th International Conference on Applications of Computer Algebra

Montreal, Kanada, 25. – 28.06.2009

<http://aca2009.etsmtl.ca/>

The ACA series of conferences is devoted to promoting the applications and development of Computer Algebra and Symbolic Computation. Topics include computer algebra and symbolic computation in engineering, the sciences, medicine, pure and applied mathematics, education, communication and computer science.

15. ICTMT 9 – The Ninth International Conference on Technology in Mathematics Teaching

Metz, France, 04. – 08.07.2009

<http://www.ictmt9.org/>

The conference will take place at the main University Campus on the banks of the river Moselle within the beautiful city of Metz in the North East of France. The site is very accessible by road, train and plane – the nearest international airports are Luxembourg and Strasbourg. Accommodation is available both on site and in nearby hotels. As well as the scientific programme, there will be a programme of social events and visits which will also be available to accompanying persons.

Papers and sessions on the following topics will be invited: integration of ICT into learning processes, technology in teacher education, designing and using Dynamic Mathematics environments, mathematics modeling with technology, communities of practice.

16. ICTMA 14 – 14th International Conference on Teaching of Mathematical Modelling and Applications

Hamburg, 27. – 31.07.2009

<http://www.ictma.net/conf/ictma14.htm>

The 14th ICTMA Conference will be held at the University of Hamburg and organized by the Faculty of Education, Working Group on Didactics of Mathematics. A variety of activities is planned, covering plenary lectures, paper presentations and working groups.

Organization: Gabriele Kaiser (Hamburg)

17. SNC 2009 – The 3rd International Workshop on Symbolic-Numeric Computation

Kyoto, Japan, 03. – 05.08.2009

<http://www.snc2009.cs.ehime-u.ac.jp/>

Symbolic and Numeric Computation (SNC) is one of the most important research areas of Computer Algebra. Previous meetings in this series include SNAP 96 held in Sophia Antipolis, France, SNC 2006 held in Xi'an, China and SNC 2007 held in London, ON, Canada.

Subjects in SNC 2009 are: Theories, Algorithms, Systems and Applications of Symbolic and Numeric Hybrid Computations.

18. CASC 2009 – 11th International Workshop on Computer Algebra in Scientific Computing

Kobe, Japan, 13. – 17.09.2009

<http://www14.in.tum.de/konferenzen/CASC2009/>

The methods of Scientific Computing play an important role in research and engineering applications in the natural and the engineering sciences. The significance and impact of computer algebra methods and computer algebra systems for scientific computing has increased considerably in recent times. Nowadays, such general-purpose computer algebra systems as Maple, Magma, Mathematica, MuPAD, Singular, CoCoA and others enable their users to solve the following three important tasks within a uniform framework: symbolic manipulation, numerical computation and visualization.

The ongoing development of such systems, including their integration and adaptation to modern software environments, puts them to the forefront in scientific computing and enables the practical solution of many complex applied problems in the domains of natural sciences and engineering.

The topics addressed in the workshop cover all the basic areas of scientific computing as they benefit from the application of computer algebra methods and software:

- exact and approximate computation
- numerical simulation using computer algebra systems
- parallel symbolic-numeric computation
- internet accessible symbolic and numeric computation
- symbolic-numeric methods for differential, differential-algebraic and difference equations
- algebraic methods in geometric modeling
- algebraic methods for nonlinear polynomial equations and inequalities

- symbolic and numerical computation in systems engineering and modeling
- algorithmic and complexity considerations in computer algebra
- computer algebra in nanotechnology
- automated reasoning in algebra and geometry

19. Gemeinsame Jahrestagung der DMV und der ÖMG 2009

Graz, 20. – 25.09.2009

<http://www.math.tugraz.at/OeMG-DMV>

Der Vorstand der Österreichischen Mathematischen Gesellschaft und die örtliche Tagungsleitung laden herzlich zum 17. Internationalen Kongress der ÖMG vom 20. bis zum 25. September 2009 nach Graz ein. Diese Tagung ist auch die Jahrestagung der Deutschen Mathematiker-Vereinigung.

Die Tagung findet an der TU Graz im Campusbereich Neue Technik statt. Das wissenschaftliche Programm beginnt am Montag, 21. September 2009 und endet am Nachmittag des 25. September 2009.

20. INFORMATIK 2009 – Jahrestagung der GI

Lübeck, 28.09. – 02.10.2009

<http://www.informatik2009.de>

Die INFORMATIK 2009, die 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI), findet vom 28. September bis zum 2. Oktober 2009 in Lübeck statt. Es erwartet Sie ein dicht gepacktes Programm von Plenarveranstaltungen,

Workshops und Tutorien, in denen Fachleute aus Wissenschaft und Praxis einen fundierten Überblick über die wichtigsten aktuellen Trends der Informatik rund um das Tagungsmotto, aber auch zu anderen spannenden Themen geben werden. Als Höhepunkt findet am 30. September der Tag der Informatik mit eingeladenen Vorträgen zum Veranstaltungsmotto statt. Die übrigen Tage werden bestimmt von einer Vielzahl von Workshops zu aktuellen Themen sowie Tutorien.

Es gibt außerdem ein attraktives Rahmenprogramm für Studierende. Dieses beinhaltet, ergänzend zum regulären Tagungsprogramm, Vorträge, Tutorien und Vorführungen zu aktuellen Themen, die sich gezielt an Studierende richten. Selbstverständlich haben alle Studierenden die Möglichkeit zur Teilnahme am wissenschaftlichen Tagungsprogramm, so dass sie sich gemäß ihren Interessen frei zwischen den Veranstaltungen beider Programmangebote entscheiden können. Darüber hinaus findet am 1. Oktober die Career Venture IT zur INFORMATIK 2009 statt. Hierbei haben Studierende die Möglichkeit zu persönlichen Gesprächen mit Vertretern von führenden Unternehmen und TOP-Managementberatungen.

Organisation:

Stefan Fischer, Erik Maehle, Rüdiger Reischuk (Lübeck)

21. Gemeinsame Jahrestagung der DMV und der GDM 2010

München, 08. – 12.03.2010

<http://www.dmv.mathematik.de/>

Im Jahr 2010 führen DMV und GDM ein weiteres Mal ihre Jahrestagung gemeinsam durch. Tagungsort ist München. Nähere Informationen finden Sie bald auf der Webseite.

Aufnahmeantrag für Mitgliedschaft in der Fachgruppe Computeralgebra

(Im folgenden jeweils Zutreffendes bitte im entsprechenden Feld ☐ ankreuzen bzw. _____ ausfüllen.)

Titel/Name: _____		Vorname: _____	
Privatadresse			
Straße/Postfach: _____			
PLZ/Ort: _____		Telefon: _____	
e-mail: _____		Telefax: _____	
Dienstanschrift			
Firma/Institution: _____			
Straße/Postfach: _____			
PLZ/Ort: _____		Telefon: _____	
e-mail: _____		Telefax: _____	
Gewünschte Postanschrift: <input type="checkbox"/> Privatadresse <input type="checkbox"/> Dienstanschrift			

1. Hiermit beantrage ich zum 1. Januar 200____ die Aufnahme als Mitglied in die Fachgruppe

Computeralgebra (CA) (bei der GI: 0.2.1).

2. Der Jahresbeitrag beträgt €7,50 bzw. €9,00. Ich ordne mich folgender Beitragsklasse zu:

- ☐ **€7,50** für Mitglieder einer der drei Trägergesellschaften
- | | |
|-------------------------------|------------------------|
| <input type="checkbox"/> GI | Mitgliedsnummer: _____ |
| <input type="checkbox"/> DMV | Mitgliedsnummer: _____ |
| <input type="checkbox"/> GAMM | Mitgliedsnummer: _____ |

Der Beitrag zur Fachgruppe Computeralgebra wird mit der Beitragsrechnung der Trägergesellschaft in Rechnung gestellt. (Bei Mitgliedschaft bei mehreren Trägergesellschaften wird dies von derjenigen durchgeführt, zu der Sie diesen Antrag schicken.) ☐ Ich habe dafür bereits eine Einzugsvollmacht erteilt. Diese wird hiermit für den Beitrag für die Fachgruppe Computeralgebra erweitert.

- ☐ **€7,50.** Ich bin aber noch nicht Mitglied einer der drei Trägergesellschaften. Deshalb beantrage ich gleichzeitig die Mitgliedschaft in der

☐ GI ☐ DMV ☐ GAMM.

und bitte um Übersendung der entsprechenden Unterlagen.

- ☐ **€9,00** für Nichtmitglieder der drei Trägergesellschaften. ☐ Gleichzeitig bitte ich um Zusendung von Informationen über die Mitgliedschaft in folgenden Gesellschaften:

☐ GI ☐ DMV ☐ GAMM.

3. Die in dieses Formular eingetragenen Angaben werden elektronisch gespeichert. Ich bin damit einverstanden, dass meine Postanschrift durch die Trägergesellschaften oder durch Dritte nach Weitergabe durch eine Trägergesellschaft wie folgt genutzt werden kann (ist nichts angekreuzt, so wird c. angenommen).

- | | |
|--|--|
| <input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/> | a. Zusendungen aller Art mit Bezug zur Informatik, Mathematik bzw. Mechanik.
b. Zusendungen durch wiss. Institutionen mit Bezug zur Informatik, Mathematik bzw. Mechanik.
c. Nur Zusendungen interner Art von GI, DMV bzw. GAMM. |
|--|--|

Ort, Datum: _____ Unterschrift: _____

Bitte senden Sie dieses Formular an:

Sprecher der Fachgruppe Computeralgebra
Prof. Dr. Wolfram Koepf
Fachbereich Mathematik/Informatik
Universität Kassel
Heinrich-Plett-Str. 40
34132 Kassel
0561-804-4207,-4646 (Fax)
koepf@mathematik.uni-kassel.de

Fachgruppenleitung Computeralgebra 2008-2011

**Sprecher,
Vertreter der DMV:**

Prof. Dr. Wolfram Koepf
Fachbereich Mathematik
Universität Kassel
Heinrich-Plett-Str. 40
34132 Kassel
0561-804-4207, -4646 (Fax)
koepf@mathematik.uni-kassel.de
<http://www.mathematik.uni-kassel.de/~koepf>

**Fachreferent Internet:**

Dr. Hans-Gert Gräbe, apl. Prof.
Institut für Informatik
Universität Leipzig
Postfach 10 09 20
04009 Leipzig
0341-97-32248
graebe@informatik.uni-leipzig.de
<http://www.informatik.uni-leipzig.de/~graebe>

**Fachexperte Physik:**

Dr. Thomas Hahn
Max-Planck-Institut für Physik
Föhringer Ring 6
80805 München
089-32354-300, -304 (Fax)
hahn@feynarts.de
<http://www.th.mppmu.mpg.de/members/hahn>

**Fachreferent Themen und Anwendungen:**

Prof. Dr. Florian Heß
Institut für Mathematik
Technische Universität Berlin
Straße des 17. Juni Nr. 136
10623 Berlin
030-314-25062, -29953 (Fax)
hess@math.tu-berlin.de
<http://www.math.tu-berlin.de/~hess>

**Fachreferent CA-Systeme und -Bibliotheken:**

Prof. Dr. Gregor Kemper
Zentrum Mathematik – M11
Technische Universität München
Boltzmannstr. 3
85748 Garching
089-289-17454, -17457 (Fax)
kemper@ma.tum.de
<http://www-m11.ma.tum.de/~kemper>

**Fachreferent CA an der Hochschule:**

Prof. Dr. Gunter Malle
Fachbereich Mathematik
Technische Universität Kaiserslautern
Gottlieb-Daimler-Straße
67663 Kaiserslautern
0631-205-2264, -3989 (Fax)
malle@mathematik.uni-kl.de
<http://www.mathematik.uni-kl.de/~malle>

**Fachreferent Schule:**

StD Dr. Jörg Meyer
Schäfertrift 16
31789 Hameln
05151-54236
J.M.Meyer@t-online.de

**Redakteur Rundbrief:**

Dr. Markus Wessler
Fakultät für Betriebswirtschaft
Fachhochschule München
Am Stadtpark 20
81243 München
089-1265-2711, -2714 (Fax)
markus.wessler@hm.edu

**Stellvertretende Sprecherin,
Fachreferentin Fachhochschulen:**

Prof. Dr. Elkedagmar Heinrich
Fachbereich Informatik
Hochschule für Technik,
Wirtschaft und Gestaltung Konstanz
78462 Konstanz
07531-206-343, -559 (Fax)
heinrich@htwg-konstanz.de
http://www.in.fh-konstanz.de/inhalte/de/KONTAKT/persseiten_nbc/heinrich.html

**Fachreferent Computational Engineering,
Vertreter der GAMM:**

Prof. Dr. Klaus Hackl
Lehrstuhl für Allgemeine Mechanik
Ruhr-Universität Bochum
Universitätsstr. 150
44780 Bochum
0234-32-26025, -14154 (Fax)
klaus.hackl@rub.de
<http://www.am.bi.ruhr-uni-bochum.de>

**Fachreferent Lehre und Didaktik:**

Prof. Dr. Hans-Wolfgang Henn
Fakultät für Mathematik
Technische Universität Dortmund
44221 Dortmund
0231-755-2939, -2948 (Fax)
henn@math.tu-dortmund.de
<http://www.wolfgang-henn.de>

**Fachexperte Industrie:**

PD Dr. Michael Hofmeister
Siemens AG
Corporate Technology
Discrete Optimization
Otto-Hahn-Ring 6
81739 München
089-636-49476, -42284 (Fax)
michael.hofmeister@siemens.com
<http://www.siemens.com>

**Fachreferent Jahr der Mathematik:**

Prof. Dr. Martin Kreuzer
Fakultät für Informatik und Mathematik
Universität Passau
Innstr. 33
94030 Passau
0851-509-3120, -3122 (Fax)
martin.kreuzer@uni-passau.de
<http://www.fim.uni-passau.de/~kreuzer>

**Fachreferent ISSAC 2010,
Vertreter der GI:**

Prof. Dr. Ernst W. Mayr
Lehrstuhl für Effiziente Algorithmen
Fakultät für Informatik
Technische Universität München
Boltzmannstraße 3
85748 Garching
089-289-17706, -17707 (Fax)
mayr@in.tum.de
<http://www.in.tum.de/~mayr/>

**Fachreferentin Publikationen und Besprechungen:**

Prof. Dr. Eva Zerz
Lehrstuhl D für Mathematik
RWTH Aachen
Templergraben 64
52062 Aachen
0241-80-94544, -92108 (Fax)
eva.zerz@math.rwth-aachen.de
<http://www.math.rwth-aachen.de/~Eva.Zerz/>